DOCUMENT RESUME

ED 290 436 IR 012 984

TITLE	The Role of the Computer in Education. Proceedings of the Annual Meeting (7th, Arlington Heights, Illinois, February 18-20, 1987).
INSTITUTION	Micro-Ideas, Glenview, IL.
PUB DATE	Feb 87
NOTE	273p.; For the proceedings of the sixth conference, see IR 012 983.
PUB TYPE	Collected Works - Conference Proceedings (021) -- Reports - Descriptive (141)
EDRS PRICE	MF01/PC11 Plus Postage.
DESCRIPTORS	*Computer Assisted Instruction; *Computer Literacy; *Computer Software; Curriculum Enrichment; Data Collection; Elementary Secondary Education; Industrial Arts; Instructional Effectiveness; Interactive Video; Learning Resources Centers; Management Information Systems; *Microcomputers; Network Analysis; Online Searching; Programing; Programing Languages; Robotics; Word Processing

ABSTRACT

The 47 papers in these proceedings describe computer technology and its many applications to the educational process. Topics discussed include computer literacy, networking, word processing, automated instructional management, computer conferencing, career information services, computer-aided drawing/design, and robotics. Programming languages such as Show and Tell and Logo are also reviewed. Additional papers discuss the use of computers to teach music, science, mathematics, reading, language arts, and physical education to normal, gifted, and handicapped students. Other educational applications included are micro-based library management, laser videodisc and computer based instruction, the creation of a successful computer program in the school, developing problem-solving skills through programming, using a microcomputer to collect eye movement data, and computer contests. (CGD)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☒ This document has been reproduced as received from the person or organization originating it.

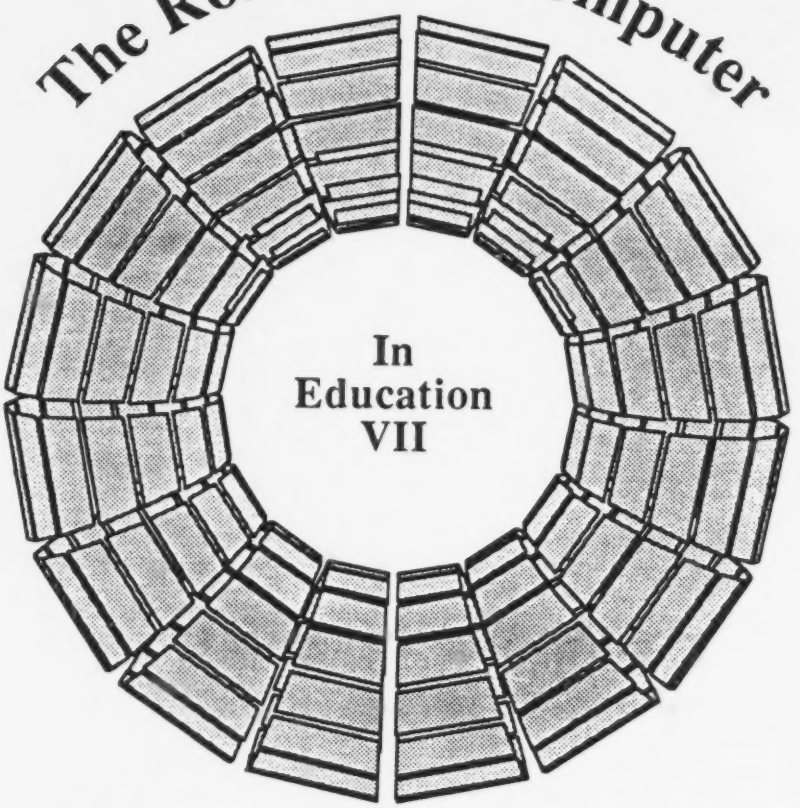
☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

ED290436

PROCEEDINGS

The Role of the Computer In Education VII



IR012984

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
Richard F. Nelson
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

BEST COPY AVAILABLE

February 18, 19, 20, 1987

MICRO-IDEAS is a division of the North Cook Educational Service Center located in Glenview, Illinois.

Dr. Lee Siudzinski, Director
Debbie Bales
Jean Dames
Kay Hicks
Richard Nelson
April Nemrava
Jean Percy
Judy Sander
Sandra Schmulbach

GOVERNING BOARD - North Cook Educational Service Center

James K. Fay, Superintendent, Elk Grove District 59, Chairman
William M. Rodman, School Board Member, Glenbrook District 225, Vice-Chairman
Leslie Lange, Principal, Morton Grove District 70, Secretary
Roderick Bickert, Superintendent, New Trier District 203
Marvin Garlich, Member at Large
Richard Martwick, Cook County Regional Superintendent
Elsie McAvoy, Reading Teacher, East Maine District 63
Wayne McKinley, Asst. Principal, Palatine District 211
Jane Rainville, Math/Computer Teacher, Evanston District 202
Donald Strong, Superintendent, Arlington Heights District 25
Thomas TenHoeve, President, Oakton Community College

Ex Officio - Administrative Agent
William J. Attea, Superintendent, Glenview District 34

North Cook Educational Service Center
2701 Central Road
Glenview, IL 60025
(312) 998-5065

PROCEEDINGS

OF

THE SEVENTH ANNUAL

THE ROLE OF THE COMPUTER IN EDUCATION

SPONSORED BY

MICRO-IDEAS
2701 CENTRAL ROAD
GLENVIEW, IL 60025
(312) 998-5065

HELD AT
THE WOODFIELD HILTON AND TOWERS
ARLINGTON HEIGHTS, IL
FEBRUARY 18, 19, & 20
1987

FORWARD

1987 MICRO-IDEAS CONFERENCE RATED A SUCCESS

I would like to take this opportunity to thank the many presenters at the MICRO-IDEAS Conference. Your contributions resulted in outstanding staff development opportunities for conference attendees.

The summative evaluation of the conference demonstrated the outstanding effectiveness of the conference. The evaluation form was sent to all 778 conference attendees. We were extremely pleased with the 60% return that resulted in a mean rating of 1.99 for overall conference quality (on a scale from 1 to 5, 1 being the highest evaluation and 5 being the lowest). We couldn't be more pleased with this outstanding rating. In addition, your many fine comments have provided us with many excellent suggestions that we will utilize to further refine and improve your conference.

As a result of the positive conference evaluation, your many fine comments, and the generous offer of the NCESC Technology Advisory Committee to help with the conference, the Governing Board unanimously agreed to have the NCESC continue sponsorship of the conference. So mark you calendar -- February 24, 25 and 26, 1988. The conference will again be held at the Woodfield Hilton, which received excellent reviews. I would encourage everyone who attended the conference to assist us in promoting next year's conference by encouraging your colleagues to attend because of the fine staff development opportunities it offers. We hope to see even more participants next year.

TABLE OF CONTENTS

How To Fund A Computer Lab With Someone Else's Money 1

Writing To Read--Perspectives After Two Years of Experience..... 4

An Introduction To Micro Based Library Management..... 10

Where Was Melvil's Head When He Ordered The Universe And They Delivered Pepperoni?..... 14

Logowriter--A Practical Approach..... 18

Technology Education At The Junior High Level--Robotics And CAD..... 22

Online Applications For The School Library..... 25

Interactive Videodisc In The Chemistry Curriculum..... 28

Ghostwriter..... 31

Instructional Goals For A High Level Programming Language..... 35

The Effect Of Word Processing On The Acquisition Of Composition Skills By Elementary Students..... 38

Interactive Tools--For A Science That Comes Alive..... 42

Search And Research--Is A Data Base A Viable Part Of The Elementary Curriculum?..... 45

Research vs. Practice In Educational Computing--A Game With No Referees..... 49

Graphing Made Easy..... 52

Monitoring Student Performance And Management Of Instruction: Mathematics, Reading, And Language Arts..... 60

"Bytes" Of Apple's B.A.S.I.C. Language Graphics And Adventure Stories (Grades 3-6)..... 63

Microcomputer-Based Information Systems For Educational Management..... 66

Beyond Software Evaluation..... 70

A Bulletin Board System For Science Teachers..... 74

Teaching Something Useful In 1000 Minutes Or Less..... 76

Introduction To Appleworks..... 81

TABLE OF CONTENTS CONTINUED

The Show And Tell Visual Programming Language 84

The Gospel According To Ackoff, Prince, Polya, Rubinstein And The Committee 93

Gifted Education Using Microcomputers 98

What Kinds Of Problem Solving Skills Do Kids Learn When They Learn Programming 101

The Illinois Career Information Systems--HORIZONS 110

Music Software--Apple IIe 114

Alpha Microcomputer Videodisc Project For Interactive Reading, Writing and Communication 125

The Evolution of the Revolution
"The Story of Driver's Education and Computer Literacy" 129

Goals and Means In Education
(Notes by a Piagetian Software Author) 139

Classification Update--Dewey 004-006 148

How To Run A Computer Problem Solving Contest 154

A Contest In Any Language 163

Instructional Computing: How Does Your District Compare? 167

TOPS: A Districuted Network For The Macintosh 172

Computers In Industrial Arts General Education 174

Exploiting The Strengths Of The Computer For Teaching Math and Reading 180

"Direct Engagement" Software For Teaching Mathematics 186

Working Toward Computerized Physical Education 195

Creating Service Levels In A Computing Information Center 199

Robotics And Logo Structures 205

Logo And Middle School Mathematics 211

Design For Decision Making 223

Collecting Eye Movement Data Using An Apple IIe
A Case Study 237

Real-Time Computer Conferencing Potential For Education 244

MIDI--Sixth Grade General Music
Marching To A New Drummer 249

HOW TO FUND A COMPUTER LAB WITH SOMEONE ELSE'S MONEY

By: Jill Alexander, Our Lady of Mercy School, Chicago, IL

SUBJECT
How to get money for a computer lab.

OBJECTIVE
To present information on how to acquire and use money to create a computer lab, without spending money from the school's budget.

MESSAGE/THEME
A school can fund a networked computer lab and computerize the administration office with spending < 8% of the total project from school funds.

I. OPENING

A. I Am Jill Alexander

1. I teach at -

2. I began in 1984 as S.S. teacher for Jr. High with the promise of being the full time CE in 1985/86

3. Prior to teaching, I worked for a foundation. I saw how well some Projects could do with a foundation's backing.

4. During 84/85, we began a fund raising program to purchase 6 computers to add to our 3 computers that were sitting idle in the library for a year.

5. Our objective was to use none of the money in the school's budget for the lab. The computer coordinator's salary was all that was to be spent from the budge.

B. Current Computerization at Our Lady of Mercy

1. Administration

a. Principal's office hardware/software
   i. 1 Apple IIe with unidisk and 128K
   ii. 1 Imagewriter
   iii. Appleworks
b. All student family information is in the database.
c. All accounting and budgets are done on the spreadsheet.
d. Class lists are computerized and easily changed year to year.
e. All the principal's correspondence is done in the word processor.

2. Computer Lab

a. 9 Apple IIe with Unidisk (64K) - 1 with 128K
(this year's fund raiser to increase all IIe's to 128K)
b. 3 color monitors and 9 monochrome
c. 2 Apple Imagewriters
d. Over 250 software disks
e. A computer education curriculum for grades K-8th.

4th through 8th

i. Keyboarding
ii. Word Processing
iii. Database
iv. Technology awareness
v. Programming - Logo

K through 3rd

i. Keyboarding
ii. Curriculum coordination

C. Rectory

1. 1 Apple IIe, unidisk and monochrome monitor.
2. Appleworks - Uses word processor mostly.

D. Tutorial Room

1. 1 Apple IIe with unidisk and monochrome monitor

This is where we are now. My objective today is to show you how we got there.

II. MY OBJECTIVE

A. Tell you how we raised money
B. Tell you how you can raise as much or more

III. GRANT SEEKERS CHECKLIST

A. Write a project description for the computer lab.

1. Be specific about your objectives and costs
2. Be sure to project a positive image for your school, the children and the world!
3. Include a budget - itemize your objectives and their respective cost.

B. Visit the Donor's Forum

1. Define the Donor's Forum

2. Publications available.

3. Use the library to identify those foundations who have historically given to schools such as yours or programs like yours, (i.e., public school vs. private vs. parochial, computer programs, reading programs, etc.).

4. Write or call each foundation for a copy of their guidelines and all necessary application forms.
    i. Try talking directly to the Executive Director
    ii. When writing, project a positive image of your school and its philosophy of education.

5. Review the guidelines and continue the process only with those foundations interested in providing assistance to educational institutions or computer programs.

6. Call the foundation again to clarify any ?'s you have about their giving program. If necessary, talk to the Executive Director again and explain your project to him/her.

7. Complete the applications. Be sure to describe your project specifically and positively. Include a list of your objectives and their respective costs.

8. Wait for acknowledgement letters. If a foundation does not acknowledge your application, call again. Simply state you are looking for verification of receipt and any information about when you could expect a decision.

9. Write thank you letters even to those who turn you down - or even call and personally thank those to whom you have spoken or the Executive Director. Do not contact those who indicate your school/project do not fit their guidelines.

10. Remain in contact with any foundation who funds your program.
    i. Invite them to come to the school
    ii. Send them pictures of the kids at work on the computers.
    iii. Let the students write to the foundation thanking them for their help to get computers. (IF POSSIBLE USE A WORD PROCESSOR).

11. MAKE THE PEOPLE AT THE FOUNDATION FEEL WONDERFUL ABOUT GIVING YOU MONEY. MAKE THEM FEEL SO GOOD ABOUT GIVING YOU THE MONEY THEY CAN'T WAIT TO DO IT AGAIN. ALWAYS PROVIDE THEM WITH THIS OPPORTUNITY!

12. Provide the granting foundations with any receipts or estimates they need.

WRITING TO READ
PERSPECTIVES AFTER TWO YEARS OF EXPERIENCE

By: Linda Allison and Dorisanne Weitermann, CCSD #59, Elk Grove Township, IL

This paper deals with the implementation and results of the first year of using Writing to Read and how the program was adjusted for the second year.

WRITING TO READ - 1985-86

The Writing to Read program was implemented in two schools in 4 first grades, October, 1985. The kindergarten program was implemented in November in one school and January 1986 in the other. The Writing to Read program requires one hour in the lab every day. As classroom teachers are already experiencing additional demands to their teaching plans; for example, Child Protection, Latch Key, etc., it was necessary to review the existing program and revise it so that Writing To Read did not become another layer of curriculum in an already busy day. At kindergarten, Writing To Read took the place of Getting Ready to Read and at first grade, the Language for Meaning program was removed. We were confident that the use of Writing to Read would more than make up for the reading readiness and language arts program.

As the year progressed and we reviewed the dated papers in the students' writing folders, it was apparent that the students knowledge of letter sounds, vocabulary, sense of wordness, sentence structure, and for some, story structure was growing weekly. For above average students in the first grade, the teachers began to see transfer of the Writing to Read phonemes to the reading groups. Ability to transfer for other students came as the year progressed. This transfer allowed teachers to shorten phonetic skill instruction time in the reading groups and gain additional teaching minutes. All of the skills presented in Language for Meaning were accomplished at the writing/word processing stations and classroom writing.

While the kindergarten teachers noted, through their informal testing, that the majority of students could recognize all their consonant sounds plus many of the vowel sounds, especially long, and that many students were writing simple sentences, their instruction in content areas was behind schedule. The emphasis for the year was on Writing to Read and mathematics. The remaining content areas, science, social studies, and health could, at times, be integrated into the writing station activities. Given this change in the curricular structure of the kindergarten program, it was important that the two kindergarten teachers know that they were supported by their principals and central office.

In making plans for the 1986-87 school year, it was decided through teacher and administrative discussions that given the two and a half hour day in kindergarten and the need to educate the whole child, the Writing to Read program would remain in first grade only. If an extended day program in kindergarten were possible, Writing to Read would be appropriate.

In an effort to retain some of the achievement gains from Writing to Read in letter sound recognition, sense of wordness, and beginning writing an approach utilizing IBM's Get Set is currently in use in six kindergarten classes. Mid-year results are quite encouraging.

WRITING TO READ IMPLEMENTATION

The success of this program is dependent on a well trained and enthusiastic staff.
In the past two years, we have been fortunate to have five classroom teachers and one central office person receive training direct from IBM. The staff members trained by IBM have then become trainers of the remaining teachers. It is important to include the principal and the technical aide in all in-service sessions. We have established four after school sessions for approximately one and a half hours each. All staff were paid for the in-service. The agenda for the sessions is as follows:

Session 1 - Chapters 1-5 - teacher's manual

It is important to hold the sessions in an already set up lab so that all equipment is available for use. This session lays the ground work for the program. IBM video could be shown at this session.

Session 2 - Chapters 6-10 - teacher's manual

This session deals with in-depth coverage of each of the stations.

Session 3 - Lead Up Week

Information received from the IBM training was used for this session. This session was held two weeks before the first teacher started in the lab.

Session 4 - IBM Stages of Writing and the Writing Process

Chapters 11-13 - teacher's manual
In addition to the teacher's manual, information from IBM training and articles written by Lucy Calkins and Donald Graves were used. This session was held about one month after the lab was in operation.

Once the labs are running, it is important to select a substitute teacher who can be used if a teacher is out ill and train that substitute while the lab is in operation. One or two full days spent in the lab as the classes cycle through should be sufficient. If this is done, there is no reason to close the lab due to illness or professional meetings. The strength of the program lies in the structure and consistency of instruction.

OPERATION OF THE PROGRAM

The first grade children were divided into four groups with approximately six children in each group. The children were given a choice as to who their lab partner would be and then the classroom teachers looked at the advantages and disadvantages of these pairings. The Writing to Read program is structured such that the children should follow a given rotation - the order of this rotation is: (1) Computer, (2) Work Journal, (3) Writing, and (4) Make Words or Listening/Library. The children work at each station for approximately 15 minutes, depending on the length of time for each new cycle word. Some words, such as uniform, take more than 15 minutes.

The lab was set up so that the Computer Station and Work Journal Station were in close proximity so that the aide could monitor both areas. The Writing Station and Make Words/Listening Library Station are monitored by the classroom teacher. Most of her time was concentrated at the Writing Station in order that she would be able to help the children develop their writing skills.

When it came to scheduling the first grades into the Computer Lab, priority was given to these classes for other special activities, such as music and gym.

Because only one class could be in at a time, we had to schedule two of our first grades in the morning and one in the afternoon, we had to readjust her normal teaching day to include both reading and mathematics in the morning--gym or music, recess, social studies/science/health were taught in the afternoon. Many topics that were covered in social studies/science/health carried over into the children's writing activities in the lab. We had to take the language arts program out of our first grade curriculum as there wasn't enough time in the school day to get to everything in, including Writing to Read. The teachers felt comfortable doing this as we felt many of the skills from the language program were being covered during the children's writing time.

After all children had gone through the ten cycles of Writing to Read, the classroom teachers, with the assistance of the aide's input, readjusted the pairings. Some of the children needed to go through all ten cycles again, some children needed to go through all ten cycles again, some children needed to go through cycles five through ten only, and some children did not need to review at all. Those children who did not need to go through any cycles again worked on IBM's Bouncy Bee program during their computer time or worked on phonics programs instead of Work Journals, and usually wanted to do more writing.

In January, the children went through a three-day training of how to work on the Magic Slate word processing program. They were given the very basic level instruction of how to create a **new** file, how to **edit** an existing file, how to **save** a file and how to make simple corrections. The children were not given any formal instruction on keyboarding as this seemed too difficult at this age.

This fall, our building had two new teachers in first grade with no knowledge or experience in Writing to Read. They took part in a two-day, IBM-sponsored workshop before the opening of school. After school in-servicing was also done at the district level during the month of September to acclimate these teachers to the program. Much time was spent by the building principal and assistant principal to help these teachers have a positive beginning when they started their children in the lab.

The week before we had the children formally start in the lab, our aide took small groups of children into the lab setting to familiarize the children with the new surroundings and to help them get used to the rotation of Computer, Work Journal, Writing, and Make Words/Listening Library. When the classes first started coming into the lab on a daily basis, the building principal or assistant principal helped the classroom teacher and aide in order to ensure that the children understood where to go each time there was a change and what to do at each station, what they were to be doing, and how to take care of the equipment. This assistance continued for about two to three weeks until the children got used to what was expected of them.

We felt that our children made tremendous growth in writing last year due to the experiences in the Writing to Read lab. Once the children became confident in their own abilities, they were very eager to write. Many of the children chose to write for a longer period of time and opted not to go to the Make Words/Listening Library station. The children were very eager to share their stories with their classmates. Many of the children wanted to know if they could also write stories at home and bring them to school to work on.

SECOND GRADE FOLLOW-UP

Now that the children are in second grade, the teachers are overwhelmed by their abilities to write and their enthusiasm for writing. The children in each of our three second grades spend approximately 30-45 minutes a day on writing. They are also using the word processing to type their stories. At the second grade level, the teachers are working with the children in a more formal way to edit their stories.

It was felt that the children received a good, solid foundation in writing in the Computer Lab during first grade. Now it is time to help them learn to expand their stories more and to do more editing.

The teachers in first grade helped the children do some minor editing towards the end of the year. It was felt that it was more important to encourage the children to express themselves and feel comfortable with writing in first grade and not emphasize editing skills. Once our second graders understood how to edit their stories and became accustom to starting off with a first draft, the teachers felt that they made the transition from the freedom in our Writing to Read program to the more formal process writing in second grade very successfully.

BUDGET/ANCILLARY COSTS

WRITING TO READ PRICE LIST* (6-85)

A typical "Writing to Read" Center could accommodate 120 students per day and would cost approximately:

	"WRITING TO READ" MATERIALS	EACH	TOTAL
3	Enhanced PCjr Color Displays and PCjr Speech Attachments	$1,363.00	$ 4,089.00
3	Sets "Writing to Read" diskettes (Set of 14) (6360700)	725.00	2,175.00
120	Student Work Journals (Set of 10) (1505638)	10.00	1,200.00
5	Teacher's Manuals (1505567)	32.00	160.00
4	Sets Instructional Cassette Tapes (Set of 15) (1505486)	125.00	500.00
2	Sets "Listening Library" Tapes (Set of 15) (1505486)	125.00	250.00
1	"Make Words" Game (1505482)	43.00	43.00
1	"Bingo" Game (1505484)	43.00	43.00
			$ 8,460.00
			- 20%
		SUBTOTAL	$ 6,768.00
	OTHER MATERIALS**		
2	Sets Children's Books	60.00	60.00
6	Typewriters	435.00	2,610.00
6	Cassette Players	25.00	150.00
18	Headphones	10.00	180.00
		SUBTOTAL	$ 3,000.00
		TOTAL	$10,248.00

The cost per student per year would be less than $35.00, based on use over five years with recurring purchase of student work journals.

* Prices estimated
** Estimated prices for materials not supplied as part of the "Writing to Read" system. Actual prices may vary.

NOTE: Speech attachment purchased separately is $300.00.

This price list is from IBM. We have made the following adjustments to their listing.

1. Typically, the classroom would be divided into four groups of 6 students. As some of our classes are over 24, we added one PCjr at the computer station. We also have a PCjr in reserve should one go down.

2. Instead of typewriters, we are using six Apple computers with the Magic Slate word processing program. This station is added to the writing table when the students are ready, generally after January.

3. Instead of the Listening Library tapes, book/cassette combinations by Bill Martin and others are being used. The pacing on the Listening Library tapes was too slow and monotone. We preferred the excellent language model of Bill Martin and the students had no difficulty following the story.

In addition to the above, further financial considerations need to be made.

1. Durable headphones and tape recorders need to be provided.

2. The location and size of the lab which allows space for each station and ease of movement of students is important. Additional tables and chairs were needed.

3. A security system for the lab is another consideration.

4. The IBM computers require jacks and splitters so two students can work on each computer.

5. Personnel costs: technical aide, in-service training costs.

6. Purchase of books/cassettes for Listening Library station.

7. Purchase materials for teacher-made and some commercial games for Make Words station.

8. Purchase of six Apple computers.

These costs need to be added to your purchase price and could equal the price of the IBM setup. This total cost is multiplied by the number of school sites to be implemented.

SUMMARY

Given the cost of the IBM Writing to Read program, Board of Education and administrative support and commitment is critical. District 59's program will include two more schools next year because of this commitment and student achievement in writing and reading.

AN INTRODUCTION TO MICRO BASED LIBRARY MANAGEMENT

By: Eric S. Anderson, Micro Libraries, Athens, OH

IN THE BEGINNING

When in the course of human events, or when your benign administrator pops in one day to inform you that you have (a) inherited some of the ancient microcomputer hardware from the computer lab, (b) have the opportunity within the next six weeks to become the district microcomputer specialist "'cause that's where all that AV stuff is stored anyway, " or (c) need to purchase, implement and have a micro based circ system up by this time next month 'cause the district down the road just put one up..... take heart, brothers and sisters, your plight is not borne by thee only. Most library automation, not just that undertaken by school library media specialists, is underplanned, underfunded, and underutilized once emplaced. Unfortunately, the pace of development has placed thee, with no training, background, or experience in the position of technomistakeola. This sad but true position is one we all find ourselves in. Yet lacking a notebook from some graduate class, forces us to buck up and do the best. At times such as these one must resort to any available course of action.

THE WIRED LIBRARIAN'S LITANY

The solution of the Wired Librarian's Newsletter has always been take it slow, learn by your mistakes, and set yourself a goal to accomplish something by a given date. There are two objects in the gospel: "Good Micro" and "Good Library." The first relies on your ability to learn what the microcomputer does well, the latter invokes current library practice in your work with this new technology. The two interweave to provide you an unshakeable background of experience, with which to judge future applications.

There are three fairly easy to master tools to begin our journey. They are present in fairly competent form in something like Appleworks: a word processor, a spreadsheet, and a data base. These tools show off the capability of the micro, and by using them you will develop your understanding of the 'how' and 'why' of the machine, at least to a level that will get you by in heavy traffic for the next ten years. If you can learn to rely on your skills in evaluating software, you can learn to ask the right questions when it comes time to do something major, rather than relying on the hype of the vendor community or the sometimes noncommittal, me-too nature of reviews.

Essential to any game plan is undertaking tasks that mean something to you. You need to identify something you do every day, for daily practice (need I mention disappointment) is the only mandate you should follow. The once a week mentality may work in aerobics, for micros it just ain't often enough. Also, since we are learning we should remember the KISS principle inherent in good advertising: KEEP IT SIMPLE, STUPID. Just as one would not tackle a cure for cancer as a first project in biology class, one should identify small tasks as their starting point for learning about the micro as a library management tool.

TASK ASSESSMENT PRINCIPLES

Once committed to learning, and with small project ideas in mind, we must then choose one of the three basic tools to put to the task. As previously mentioned we have three: word processing, spreadsheet, and data base. Debate may be heated, and blood may be shed over the choice of the "right" tool for the task, but in developing your own micro sense you will also be able to reach the second level: deciding what software does the job you want done.

To start you on your trek, each of the tools has a specific strength. For purposes of discussion, take the primo function, identify characteristics, and apply it to a task. In reflection we may choose another tool, or set up the file in another manner, but that is the level of consciousness we are striving for. Word Processing, as its rather harmless title would reveal, has as a strength the ability to change and manipulate text easily. Spreadsheet, which sounds like a nasty motel on the other side of town, has a backbone founded in mathematics and "number crunching". Data base reminds all of the old joke: "Know yourself. When in doubt call the FBI." For purposes of introduction the concept of an electronic file cabinet will suffice.

WORD PROCESSING

A word may be a word may be a word, but libraries spend a great deal of time and effort producing an array of documents. These vary in scope and sequence from reports to managing authorities, documents designed to help a user find items or understand library practice, and include the array of documents necessary to keep the place running - policy and procedure manuals - just to name a few.

I would wage some very high stakes on the fact that many of the folks in this room do not have current, up-to-date selection policies. The problem is that such policies, once they are finally typed, usually do not get changed as often as we would like because the typing (or final presentation if you will) requires more effort than the thought. Word processing has as a principal strength the ability to insert or delete blocks of text into a previously created document. Once the changes are made on the screen, before you ever commit them to paper, the micro takes over the laborious duty - getting the words onto a piece of paper.

Any library management chore that requires updating, and is primarily text in nature becomes a candidate for word processing. The hidden agenda of this tool is the "screen editor" or the ability software gives you in moving the cursor anywhere on the screen and doing virtually anything you want. As you begin to evaluate software, you begin to think - " Wait a minute, my word processor is a lot easier to use than the screen editor in this program." The problem is not in the micro, it is in the software that is controlling the micro. Your screen editor in any application should be as good as the screen editor in your word processor. At least the screen editor should approach a word processor in an application program.

Word processors generally don't do math (much in the same way I don't do windows) and for the most part they don't sort or manipulate the information in their files. If you want it to look pretty, you use a word processor.

SPREADSHEET

The hand has been tipped that a Spreadsheet is a number cruncher. This type of software divides the screens into "cells" which are usually referred to as 'rows' and 'columns'. The spreadsheet allows you to manipulate the contents of a cell, or a group of cells, against another cell or group of cells.

In many ways the spreadsheet is the most difficult piece of software to learn to use, primarily because it is contorted and exacting. The formula that controls a numeric contortion cannot know whether it is inherently wrong or right, that test is left up to the operator - or in our example - left up to you.

What type of number activities are prevalent in libraries? Often we are required to provide statistics to a governing authority. Most of us have budgets to manage and orders to generate. Many of them are perfect candidates for a spreadsheet application. Consider the annual supply order I used to put together every June. All year long slips of paper would congregate, decrying the need for catalog cards and printer ribbons and a plethora of little stuff we all need in order to run our shops. The trick was always matching the stuff I needed with a budget figure. In the paper, calculator and eraser days I could barely type the order from the worksheet after the machinations of removing three of twenty-four bookends from the order to get it under budget. With a spreadsheet you merely set up columns for number wanted, price each, a calculation to figure total item cost and a cell adding the item costs together. Getting it under budget was as easy as having the computer type the look-alike requisition "ship as per attached list" and stapling the spreadsheet printout to it.

Spreadsheets for the most part have really limited screen editors, and in most cases do not sort. If it's numbers you want to crunch, you normally want to use a spreadsheet.

DATA BASE

Earlier we intoned that a data base was an electronic file cabinet. The cabinet is called a "file." The folders normally used have "record" electronic counterparts. Items or data within the folder go by the moniker of "fields." Data bases are files made up of records which are composed of fields. If you want to manipulate any of the data, normally each element has it's own field. To create a file where you sort by last name, then it must have a field for name where the last name is entered first, or you must have a field for last name and a field for first name.

Data bases are great sorters, but their math capability is usually limited to totals, averages and counts. Their screen editors are also usually very weak. Hard copy printouts also often end up looking like your phone bill. If you want to see the same set of information displayed (or printed) in a variety of ways, then our friend the data base is your tool.

LIBRARY UTILITY SOFTWARE

Up until this point, we have limited our discussion to three basic tools. There is a variety of task specific software that is referred to as "utility" software. Unlike generic utility programs that crack copy protection schemes or sort directories, library utility software is designed for single function application. Most popular in this category are bibliography and catalog card production software, although there are many more types available. This utility software is designed as a shortcut to the hassle of designing a template (blank form) for your data base or spreadsheet to tackle a task.

In general, this utility software is a "canned" data base application. The author has decided what fields to include, how long they will be, and what print options the user will have. The variety of this software merely demonstrates how different folks tackle the same problem differently. If you have developed your own skills, that is if you understand what "good micro" and "good library" is, then you have the ability to evaluate how well the screen editor functions, what sort of math it performs, and whether or not it sorts the way you want it to sort. If you have knowledge then you can evaluate. If not, you are left to the mercy of the vendor. Hey folks, they are still selling card production systems like mad that don't use AACR2 and there is a very popular circ system that still uses 40 column screens! Their user population neither knows enough nor cares enough to scream for our Wired Librarian's Litany. May St. Jerome be with them.

THE RIGHTS OF PASSAGE

As you begin to implement the micro into the management scheme of your library, you develop a thought pattern. Is it text? Do I need to crunch numbers? Is this a sorting problem? As this mentality sinks in, you will realize that even though a bibliography is a sorting problem, the printouts aren't acceptable so you start looking for a bibliography program that fits your needs. When you finally decide to tackle circulation, you hope that the program you have been using for card production writes files that can be "dumped" into your new project. If you've developed your own expertise, you more than likely will be more than half the way home. If you have relied on vendor hype, hope someone else is watching out for you.

WHERE WAS MELVIL'S HEAD WHEN HE ORDERED THE UNIVERSE AND THEY DELIVERED PEPPERONI?

By: Eric S. Anderson, Micro Libraries, Athens, OH

It has become apparent in conversation and in proposals I have received, that the dichotomy in librarians use of technology in general and, specifically micro-computers, I have feared for so long is not only raising it's head, but waving it's flag. My concern is simple. Technology is not the issue. Rather our ability to understand the technology and the way, shape and form it can change our way of doing business should be our focus and direction. A faster processor, new storage devices, user interfaces are interesting to a minimal degree to a minimal audience. You should know enough to know that you don't know, to paraphrase the ancients. What is inside the box for the most part is insignificant. Where the box is placed is perhaps slightly more important. How the box is used and what it can do for the people that use your library should rise unceasingly to the forefront. What it can do is more important than the color or the shape or the gigabytes.

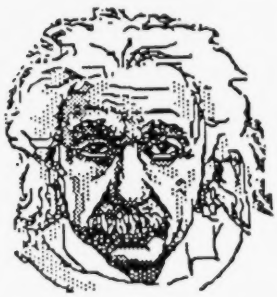
The library profession has spent a great deal of time, effort, and need I say money, taking pieces of paper and creating bit images. Copies of what, how, and when transported from the way we have always done things to an electronic environment. Mirrors with chips and keys that go hum in the night. I use this quotation in workshops frequently, and am upset because I cannot locate the source:

> Librarians have always been resistant to change. When the manual typewriter invaded libraries, librarians said it was more trouble than it was worth. It was too difficult and awkward to hoist the typewriter up to dip it's corner into the inkwell.

We are unable to disassociate ourselves from the past. We continue to do business primarily the same way we have always done business with the latest flash and fancy thrown in for good measure. Rarely do we step back and say, "What are we doing, why are we doing it, and will we continue to do it?" Constantly library automation is "doing the old stuff better" - with better assumed cheaper / faster / easier.

I do not profess to be a student of library history, but I can safely say that Melvil Dewey and his enduring system to categorize the world into ten ranges with ten ranges ranges inside that shell with a myriad of layers, used primarily at the discretion of the local library, is both a blessed and blamed example of how libraries order their universe. Change the schedules at predetermined times, in a predetermined manner with a predetermined purpose - progress. The intent, again from a users perspective, is to put like items together. Every study I have seen comes to the same conclusion: the catalog leads the user to the shelf and the shelf is where the real selection occurs. The shelf is organized by the derivative of Melvil's world or some other scheme. Economy of size dictates the efficiency and, therefore, the schedule used orders the users universe.

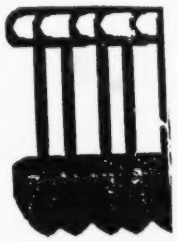
The assumption always being the wisdom is in the schedule.
The link being the physical presence ordered by the schedule.



The ultimate goal and the particular problem is similar to a Moebious strip. One turn, one twist, and the surface our mind tells us has two sides actually has one. This fashionation with turning in on ourselves not only limits the advance or improvement of service we provide, but a generic inbreeding that gives VDT the ability to wipe out the librarian strain without a blink.

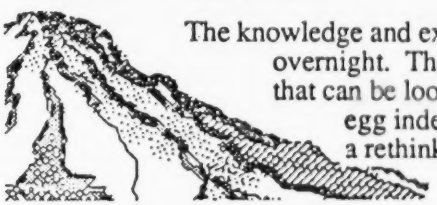
Down the hall from me works a consultant whose idea of the automation world is 180 degrees from mine. He spends his time writing drivers and producing exec files so that the operators, the lowly paid help doing the actual work, can turn the machine on and "do it." In one sense he wants the user to know no more about the machine and what it can do than the limits he sets. In certain environments this is fine and dandy. In others it is totally inappropriate. For anyone who will use the tool to extend their creativity in doing their work, it is inexcusable.

Allow me to present a seemingly mundane example. It you want to dig a hole, you go out and find a shovel. It doesn't take much training or experience to operate a shovel. If you want to dig a great number of holes, you either go out and acquire a great number of shovels and their associated operators or you find an alternative. Whatever alternative selected, the shovel mentality has as its limit the strength and agility of the shovel operator. If you dig one small hole a year then an adequate shovel and an adequate operator (nay say oneself) suffices.



In a single task environment, the canned solution is quite acceptable. This is demonstrated by the great number of micro based circ systems put up during the last several years. Recently I had the pleasure of attending a large library gathering where competing vendors sponsored "users group" meetings. At one, the discussion is "how is so and so in (insert any state you care)?" At the other the attendees wanted to know when certain enhancements would be available, to what degree would statistics files be transferable to another software package, and when were they going to add a ROM interface? The first vendors package is still state of the art 1982; using 40 column screens, slow sorts, and relatively cumbersome storage techniques. You're right that the users of that system didn't want to talk tech, be excited about another gig of RAM, or even care what a SCSI port would do for them. They have their shovel in their hand and they are digging one hole a year, and thinking it's a decent hole.

It is my contention that we are making an extremely fatal flaw in the implementation of computers into libraries. We spend so much time taking paper files and letting the computer handle it that we either forget, or do not understand, the nature of the machine. When I look to automate functions, I ask instead, "what can the hardware/software do and how can it do library stuff." That is far from exec files and drivers. Need we not ask, "Why should we dig a hole?"



The knowledge and experience necessary to reach this position do not come overnight. They are the results of a series of successes and failures that can be loosely called "experience." This experience is a speckled egg indeed, but in breaking its shell it gives birth to a change, a rethinking of library automation.

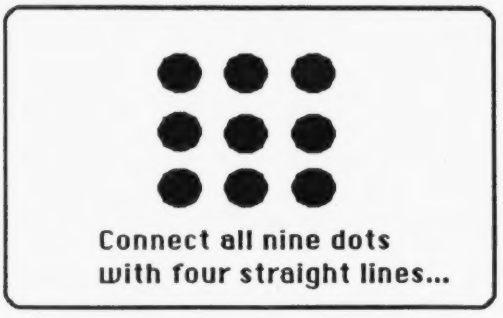
To this point in the history of the micro and the library, we have been burdened by one undeniable fact: what we need to store and retrieve is far too large for even the largest and fastest hard disk drives available. MARC records are just too big; the indices needed to retrieve data are just too big; we have always had half a loaf.

Another library intense computer use has also caused us to dream up Mickey Mouse patches and expensive alter ordered. Because of the economy of scale, it is too expensive to load all of the up-to-date, current, if you will, databases other than on a single machine and allow folks to access them over phone lines. 212 mentality I say, Bell 212 to be exact.

Furthermore our access to many of these tools is also limited to the points or hooks provided by the vendor. Even with extensive annotations, we are not always sure that the citation in hand contains the information our users need. We opt for offline printouts or interlibrary loan; both of which fail to meet the principle test of library service -- providing what the user wants when they want it. We stall and delay and increase everybody's cost by borrowing it from someone else because that is the only way we have to obtain it.

All of this is changing. The evolution of technology has provided us with a tool that will store, index and retrieve many hundred megabytes of information - the CD ROM. The current generation holds some 500 meg; perhaps the next generation will hold 10 or 100 times that much. Whatever the standard is today, in my opinion, is insignificant. What is significant, is that the technology is beginning to be adapted to libraries, and the development process will bring to it more and more applications.

On my desk at the Sioux City Public Library, I had a CD-ROM unit hooked to a micro. On this disk are some 470,000 bibliographic records with some 1.1 million holdings symbols. Instead of relying on printed indices, or expensive bibliographic utility, I fire that puppy up and search my interlibrary loan requests. I don't find all of them, nor would I expect to. The cost, the micro and the ROM hardware, which I need not remind you are one time investments, and the ROM itself, which the State Library spent about $70,000 developing and distributing. The next one should cost less than half that, and the one after that half of its predecessor. Compare it to the investment in the utility and its twelve cents per display (with an average of three per search) and you can see the economy of the ROM on 80,000 searches per year. Yes, the ROM we currently use has problems (for one it doesn't write disk files or print ALA mail forms for libraries not on the statewide network), but that will come in time. The difference is the project was conceived with "what will the technology do and how can it be applied to interloan? We took the capabilities and designed a product.



If you could come away with one idea from today's gathering, I would hope it would be that by limiting yourself to transcribing paper to bits you are creating artificial barriers that need not exist. Know, and examine what the technology is capable of doing and be vocal enough to ask the vendor community to produce tools for you to use. If it is a full text searchable version of Indexus Medicus, go for it. If it is anything less, you are letting the old ways get in the way of new technologies.

You can take the wheels off the cart if the cart floats on a cushion of air.

LOGOWRITER - A PRACTICAL APPROACH

By: Darlene Bandomir, Cherokee School, Lake Forest, IL

LogoWriter is an exciting way to bring Logo and word processing together giving users greater diversity and more creative options. Children enjoy using it and develop many new skills as they experiment. Educators can easily implement LogoWriter into their classrooms, even if their experience with computers is limited.

The most exciting aspect of Logowriter is the way graphics and text can be combined. Students learn basic word processing skills without making the process too involved. With graphics creativity is added to writing. Children want to write more. Even reluctant writers become successful and find out that writing can be fun.

There are a great number of commands for children to learn while working with LogoWriter. To help them as they work at the computer, commands can be written on sentence strips and displayed where they can easily be seen. As new commands are taught they can be added.

DRAWING

The possibilities seem to be infinite when using any Logo program for drawing. When using LogoWriter, children are able to see the turtle's direction more easily since the turtle has legs and a head. Geometry, problem solving, and creative expression are all put to the test with this component of the program.

The activities that follow are just a few of the possibilities for projects in the classroom. A large compass is helpful to students as they learn about angles and work on these projects.

1. Racecourses - Make screen size copies of the racecourses provided. Have the students place the turtle at the starting line and work their way through the course. Students could be timed with a stop watch and given awards for the fastest times.

This activity can be used after the students have some understanding of BK, FD, RT and LT.

2. Define procedures for squares, triangles, rectangles, circles and stars. Once the students have some understanding of these procedures, they can try some problem solving using the following designs.

PROCEDURES

These procedures should be put in the Flipside of the page.

```
TO SQUARE :SIZE
REPEAT 4 [ FD :SIZE RT 90 ]
END
TO RECTANGLE :SIDE1 :SIDE2
REPEAT 2 [FD :SIDE 1 RT 90 FD :SIDE 2 RT 90]
END
TO TRIANGLE :SIZE
REPEAT 3 [FD :SIZE RT 120]
END
```

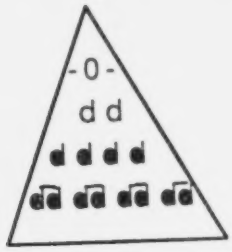
```
TO CIRCLE :SIZE
REPEAT 36 [FD :SIZE RT 10]
END
TO STAR :SIZE
REPEAT 5 [FD :SIZE RT 144]
END
```

MUSIC

Music can easily be programmed into the flip side of Logowriter. Children find this component challenging, but are willing to work hard. There are some basic skills needed before they begin. Charts help students as they learn to work with the notes, duration and rhythm.

The following ideas will help children get started. LOGO TURTLE and the TURTLE SONG are good ways to show children some of the things the turtle can do.

RHYTHM PYRAMID



NUMBER VALUES

-0- = 80
d = 40
♩ = 20
♪ = 10

Two resources for music and Logo are:

TEACHING and COMPUTERS, March 1986
(Let's Make Music! by Dee Miller)

TEACHING and COMPUTERS, September 1986
(Turtle Song and Dance)

Music Note Values

HI	C	520	C#		278	B		492
C		262	Bb		466	HI	G	704
A		440	G#		416	HI	E	640
G		392	HI	Eb	608	F#		370
HI	D	578	F		350	HI	C#	548
E		330	HI	C	520	Eb		312
D		294						

GAMES AND CENTER IDEAS

There are many Logo skills that can be taught off the computer. The following ideas have helped elementary students learn more about Logo.

1.) Direction Game

This game will reinforce the concepts of clear screen, show turtle, left, right, forward, and backward. You will need to make a grid on a 10" x 12" piece of poster board. Draw half inch squares across the board leaving a one inch border. Draw darker lines at the center point in the X axis and the Y axis. Where the two lines meet, draw a turtle. Next make two dice out of half-pint milk cartons. One once die put the commands FD and BK in red, ST and CS in black, and RT 90 and LT 90 in blue. On the other die write the numbers from one to six.

Children will need different colored markers. Put eyes on the front so the child will know which direction the turtle is facing. A red chip will also be needed; this will be placed at some random point on the game board.

To play, the children take turns rolling the dice. They must roll a ST to get onto the board. As they move they try to be the first one to get to the space where the red chip is. If they roll CS they must take their piece of the screen and begin again.

(This idea is adapted from TEACHING and COMPUTERS - January 1987-Micro Ideas)

2.) Computer - Turtle March

Using pictures of turtles and computers, the children will match terms with definitions. These can be kept in a large envelope. The children can check their answers with a key included in the envelope. PRINTSHOP shapes work well for this.

3.) Flashcards

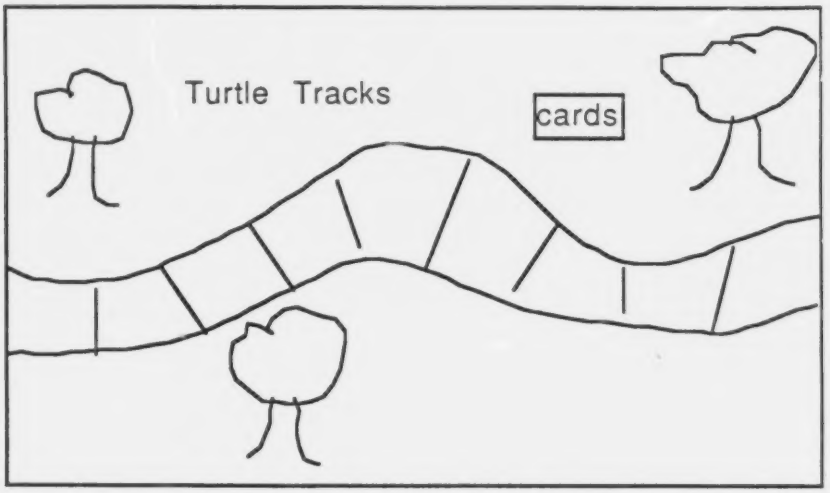
Flashcards are helpful when learning terms. Use the terms from the LogoWriter Language page. Pictures of turtles make the cards more interesting.

These can also be used for a matching game if a definition is on one and a command on the other. (Concentration)

4.) Turtle Tracks Game

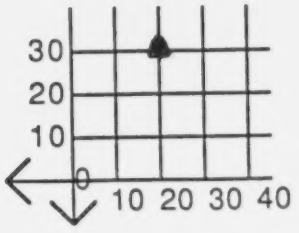
Make a game board with a turtle trying to find its way out of a maze. The board should have at least 20 spaces.

Cards will allow the children to move forward and backward, as well as answering questions before they roll a die. If they answer the question correctly they go forward, if not, they go backward the number of spaces they have rolled.



5.) Center Task Cards and Problem Solving

Laminate task cards with problems on them. The children can attempt solutions and then use the computer to test them. The designs from the drawing pages without solutions work well. Simple grids with a turtle and a Logo command also work well. The child carries out the command and then checks her work on the back.



BK 20

RT 90

FD 10

TECHNOLOGY EDUCATION AT THE JUNIOR HIGH LEVEL
ROBOTICS AND CAD

By: Richard J. Banducci, CCSD #59, Arlington Heights, IL

INTRODUCTION

One of the most interesting challenges faced by American educators today is the challenge of "Technology Literacy". Industrial Technology teachers in the past have faced many and varied curriculum changes; i.e. metric education, integrated arts, the world of construction, the world of manufacturing, mainstreaming and creativity problem solving, just to name a few. Some of these changes have left modest shifts in our curriculum, however, I believe the Technology Literacy emphasis we are now facing is one that will be with us for a very long time and have an enormous affect on our Industrial Education curriculum. What I am going to do in this paper is to present a brief statement regarding what seems to be a fair definition of Technology Education, and explain a model program as seen through the eyes of a professional educator. After which I will lay out the specific program I am involved in, and explain how this program fits into the model presented.

DEFINITION OF TECHNOLOGY EDUCATION

Through conversations with other educators, representatives from industry and appropriate literature the following definition of Technology Literacy seems reasonable: "Technology Literacy is the ability to understand, communicate and apply technical information to the solution of problems." Other phrases which seem to fall nicely into this area are: "to ensure intelligent decision-making" concerning future course selection as well as career opportunities; "To allow persons to function more effectively in an increasingly technical society and workplace." Finally educators may find the following statement of behavioral changes as indicated by Dr. John R. Lindbeck, Director of the Institute of Technological Studies at Western Michigan University, helpful in the re-direction of their industrial education curriculum:

A technology literate person will be better able to:

1. participate in technical progress
2. evaluate technical progress
3. monitor and control technical progress
4. adapt to changes resulting from technical progress
5. understand how technical progress affects the production of goods and services
6. apply appropriate elements of technical progress to the workplace

Towards the achievement of a successful Technology Literacy program, Dr. Lindbeck cites three available options:

Option I: **Visual Media**
This is simply an enrichment version of a traditional industrial arts program. In this option the teacher would make use of filmstrips, slides, movie presentations, charts, posters, etc., to familiarize the students visually with "high" technology" concepts.

Option II: Interactive Software

This option would allow for the use of computer software, such as a beginning level CAD (computer aided design) program to be used in a drafting and design class this type of software, coupled with other industrial technology software can be used to familiarize students with techniques and terminologies being used in industry today.

Option III: Advanced Technology Lab

The third option is a curriculum modification which allows a facility to be established, such as a technology lab. In this self-contained lab, students will experience hands-on activities in such advanced technologies as robotics, hydraulics, pneumatics, electronics, CAD, CAM (computer aided manufacturing) laser technology, fiber optics and any other area in the scope of "new technology" as dictated by the range and applicability of the instructor, his/her qualifications and financial resources.

Through these examples one can get a much clearer picture of what a successful Technology Literacy program might entail. This type of program is suitable for middle, junior high school and senior high school industrial education curriculums. Of course to achieve the completeness of a technically literate individual, cross reference as well as elements of technical progress must be taught and re-enforced in other areas of the curriculum--science, social studies and home economics must also contribute. Although these contributions can be achieved in a number of different ways, some form of technology reference is vital to the total Technology Literacy program.

The specific Technology Education program I involved in has indeed gone through the three stages as indicated by Dr. Lindbeck. We first began approximately three years ago with some slides and software related to CAD and Robotics. After teaching a three week pilot course on Robotics and CAD during a summer session, the interest levels on the part of the students as well as the instructors began to grow. The following school year after a presentation to the school board, we were able to obtain enough equipment to formulate one Technology Lab. This lab was a "traveling" lab which spent twelve weeks in each of our three junior high schools. Once again after the successful completion of this pilot program, we purchased equipment to establish a technology lab in each of junior high schools.

Each lab is equipped with the following basic items:

3 computer stations (monitor, computer, double disk drive & printer)
1 pneumatically activated Robotic Arm
1 electronics kit
2 pneumatic training kits
2 hydraulic training kits
2 mechanical training kits

In addition to this hardware, each lab has an assortment of related computer software. Such titles as Gears, Factory, Super-Factory, CADDRAW, Pen Painter, Robcat, etc., are amongst the library in each lab.

Students will experience hands-on activities in areas such as CAD, pneumatics, hydraulics, mechanisms, and learn how these technologies are being used in industry today. They are also given an opportunity for advanced course work in an Advanced Industrial Arts program for 8th graders. As with any implementation of new curriculum, effective leadership is essential. Successful leadership at any level must demonstrate the following characteristics:

1. **Vision** - effective leaders must have a vision of what they want to achieve, and even more importantly, they must be capable of conveying that vision to their staff.

2. **Staff Involvement** - effective leaders involve staff in decision-making, when appropriate.

3. **Support Curriculum and Instruction** - creative and effective leaders promote effective curriculum and instruction.

4. **Monitor** - effective leaders monitor progress - not only students, but teachers also.

5. **Resourceful** - effective leaders do not take "no" for an answer. They have the ability to get what they are after.

The implementation of Technology Education is an ongoing process, constant updating, classroom visitations, staff development and support are all vital areas for the success of Technology Education.

BIBLIOGRAPHY

U.S. Department of Education, What Works - Research About Teaching and Learning, U.S. Dept. of Education, 1986.

ACIATE, American Council on Industrial Arts Teacher Education, The Dynamics of Creative Leadership for Industrial Arts Education, McKnight Publishing Company, Bloomington, IL, 1983.

Lindbeck, Dr. John R., "Implementing Technology Literacy in Your Classroom", Hearlihy Times, Fall, 1986.

ONLINE APPLICATIONS FOR THE SCHOOL LIBRARY

By: Eileen Bell, Township High School District 214, Mt. Prospect, IL

Adding a modem and telephone line to the computer in your library can result in many changes in the services you perform, as well as the methods that you use to perform them. You can also expect to see a difference in the way you are perceived by the users of your services. This is the third year that our high school library has been online. We have grown considerably in our capabilities, and are planning for increased usage in the future. One of the primary goals in using online connections is to increase the kinds and numbers of resources which we can make available to our students and staff. We have a limited budget, but we can offer so much more than we could ever hope to purchase. Whether we are seeking book material or article information, our online capabilities give us sources, and our interlibrary loan network gives us the actual documents. Besides the obvious advantage of giving research material to users, we also feel we are teaching students that our library, as well as all other libraries, is part of a larger world of information which they can access at anytime for any information need.

Another primary reason that we are providing information via computer is to teach our students the skills that they will need for the future. More and more information is becoming accessible by computer. More college and public libraries are computerizing their card catalogs. Our students need to know how to use a computer to do their research, and we will be remiss in our duties as instructors of library skills if we do not address the issue of training students for the library of the future. We can't still be teaching high school students to use a card catalog when their local public libraries and, most likely, their college libraries won't even have a card catalog. The skills needed to adequately use a computer to do research are complicated and need a progression of instruction and practice. We need to begin planning and implementing this kind of instruction into our curriculum.

Once a decision is made to tap into online resources, the next task is to gather the equipment necessary to make connections. Either a microcomputer or a dumb terminal can be used. Terminals would be less expensive and would not require modems as they have acoustic couplers or built in telephone line connections, but they would probably run at 300 baud which is a slower speed. Connecting at 1200 baud is more desirable because almost all online services charge you by the amount of time you are actually connected to their system, and the faster your equipment processes the session, the sooner you can disconnect. Use of a microcomputer allows more flexibility than a terminal, though it also requires modem, printer, and communication software. Some software allows you to upload and download searches so that you can edit before or after transmitting.

The other equipment needed besides the computer is a telephone line, which should be a direct outside line and not one that goes through a switchboard. The type of telephone service should accommodate calls to your farthest point. Telecommunications services like Telenet and Tymnet have many numbers available from across the state of Illinois.

Now that you have your equipment and are plugged in, what different sources can you call? The first of our online connections was to Dialog. In 1981, our district decided that online database searching would provide a valuable source of information for research projects and for the reference needs of faculty and administrators. We expanded to include BRS the following year, and now, between these two vendors, we have access to over 300 databases of information which cover almost any topic you can name. Most of these databases are bibliographic and are searched as a means of supplying a bibliography of sources. Information of this type may be used by teachers who are adding a new segment

to a course, writing a paper for a class they are taking, designing a new course for the curriculum, upgrading their teaching methods, or engaging in any number of other research activities which require information that the school library could not possibly supply. The research needs of our staff should be as important as those of our students, and school libraries are not adequately equipped or budgeted to handle many of these research needs. Being able to escape the confines of your four walls and tap into the greater resources which are available is not only a valuable asset, but, perhaps, should become the expected norm.

A library budget is very confining in regards to the wealth of information which is available. Paying for information which is actually sought should also be weighed against paying for books that might, or might not, ever be used. The availability of online access to computerized versions of printed sources may even alter decisions on what books will be purchased in the future. If we took the money now spent for the New York Times Index, and spent it on computer time to the National Newspaper Index, I would guess that we not only would have money left over at the end of the year, but that we would have had many more successful searches in the index because of the ease and flexibility of computer searches.

Online database searching is a major use of our online computer and gains the most attention since it involves money, staff training, and documentation. There are, however, other places to call with your modem. One such place is another library's card catalog. We are very fortunate to be served by a public library, which not only has an online card catalog, but one which makes the catalog available to other computer callers. We can dial right into the computerized catalog, perform author, title, or subject searches, and receive an immediate printout of books available at that library. If your local public library does not have this capability, there are other library catalogs you can access, such as Northwestern University Library, or LCS, a group of university libraries around the state which includes the University of Illinois. We use these sources often for students who need resources beyond what is found in our card catalog. Our library systems do a wonderful job of providing materials for us on interlibrary loan, but the hard part for us is locating what materials are available. Online connections can easily put you in touch with an enormous volume of new sources, and the best part is that there are no online charges or fees other than your telephone bill.

In our library, we are trying to speed up the delivery of documents ordered through the Central Serials Service, because our students are perpetually late in making requests. After they have put in the time and effort of creating and implementing a search strategy, we want to be sure that the articles they request arrive in time to be actually used. One way that we can save a few days is to request our Central Serials articles directly online. Central Serials has the necessary instructions for those wanting to make a request by computer, and, once connected, you simply type your request in directly. This connection has no online expenses other than the local phone call.

We can use the resources of online databases in support of acquisition and cataloging activities in the library. There are several book databases, including Books in Print, which would assist in locating information for ordering titles. We like the convenience of having the printed index on hand, but actually, considering the number of times we use it and the cost of it, I am sure it would be much cheaper to devote the money to online time. We would then also be able to search Forthcoming Books and Paperbound Books in Print, which we don't buy.

Cataloging tasks can be assisted by use of DIALOG's LC Marc and Remarc databases which collectively give access to all Library of Congress cataloging since 1897. BRS has the OCLC database which is designed for subject access to OCLC records, but also includes all the information a cataloger might need. At the present time, we use these databases primarily as a backup to our regular cataloging information sources.

Library instruction for students in our school is going through a major change. Last year we began teaching online database searching to junior and senior students in Expository Writing II classes. The classroom instruction focuses on an explanation of what databases are, what results can be expected when searching them, what Boolean Logic is, and how to construct a search strategy. Students are not required to do a search when researching their term papers, but they are invited to consult with the librarians and perform searches if they so desire, and if their topic lends itself to a computer search.

This year we have expanded our instruction program. We now have a second computer station in the library, in addition to the one we previously had in the library office. This second computer is always available for students to dial into library card catalogs, or with the assistance of a librarian, to perform database searches. Our instruction program for database searching is expanding to all research classes in the curriculum, and our hope is to eventually have given the instruction to every student before he graduates. Besides the database searching instruction for research classes, however, we are using all of our class instruction opportunities to include basic instruction on using the computer to access other card catalogs. We want students to know that sources of information on their topics can be found in our library even if we can't purchase and house the material. We are part of a larger library network, and we can get for them almost anything they request.

If you agree that teaching computer searching skills is indeed a worthwhile activity, but don't yet have the ability to get online, there are a few alternatives you might consider. One such program is titled Information Connection, and it can be used as a practice and training program for performing online activities. It also will actually dial, and it becomes a communication software package if you do get hooked to a modem and telephone line. Another self-contained program which can be used to teach Boolean Logic and elementary searching skills is the Eric Microsearch. These disks contain a small subset of Eric records that a student can practice retrieving with a variety of search tactics. Neither of these programs provide the flexibility or require the skill needed for true online searching. However, if online connections are not available, they could be used as an introductory activity.

We have been pleased with the results of the online activities at our library. Our goals in presenting these capabilities to our students and staff have been to provide the resources for today, and to teach the skills for tomorrow. We have been rewarded by very enthusiastic users. Even those who are not yet ready to try the computer access have been receptive to the instruction, and have acknowledged the need to be informed of these research methods. We plan to continue and expand our offerings, and to add to the numbers of, not only informed, but truly experienced users.

INTERACTIVE VIDEODISC IN THE CHEMISTRY CURRICULUM

By: David Bergandine, University High, Urbana, IL

The recent development of laser video technology has revolutionized computer-based instruction. Educationists have attempted to exploit the computer in every way to suit it for a variety of instructional purposes in a multitude of disciplines. Included in this would be the creation of programs and lessons of an interactive nature.

However, one major facet of computer-based instruction that has not kept pace with advancements in lesson design and programming is that of graphics. Certainly most CBI programs do include graphics of some sort - though they are frequently no more elaborate than stick figures or symbols. While the creation of realistic images requires expensive graphics hardware, the incorporation of live video is now possible via the interface of computer with a laser videodisc.

The laser videodisc is perfectly suited for marriage to the personal computer. In just seconds the computer can access any of over 50,000 frames. Video images can then be ordered in a nearly endless number of combinations - thus giving rise, not only to interactive lessons in the above sense, but also to a personalized sequence of live video. In the chemistry curriculum, this amounts to a student performing his or her own laboratory experiment and obtaining results that differ from any other student.

For those well familiar with CBI, the appearance of live video on the screen has almost a magical quality, but after the initial impression, it seems like the most natural partnership. In this particular set of lessons[1] the video is used in three ways: 1) as live or still video alone on screen, 2) as video with computer generated text or graphics overlayed, 3) video partitioned in boxes or windows produced by a process of painting a computer image to be situated over the frame.

The techniques that have made these programs a success grew out of the combined expertise of a renowned CBI author and a seasoned veteran of videotape production. In order to depict the effectiveness of lesson design and content, and to convey the impact upon student viewers, it is instructive to turn now to the students impressions and comments.

Following a twenty five minute session working two lessons involving gas laws, students were asked to summarize in one to two pages what they had learned and their perceptions of interactive videodisc instruction. These excerpts represent the opinions of most students.

"The points that were made in the program were clearly stated and easy to follow: It made things a lot clearer - even things I already know. After using the laser disc, I felt much more comfortable with the subject and more confident with my answers."

Prior to this lesson, there had been no instructional time devoted to the material, and there had also been no related assigned reading. On the average, students could correctly answer less than half of the questions on a simple quiz over the lessons' content, but after treatment, responses improved dramatically. Their discussion of the material on the in-class essays was quite impressive considering they had only a brief exposure to the topic.

"I could go at my own pace and make sure I understood the material. It's a quick way of learning stuff."

These two divergent comments illustrate that like any good interactive lesson, the program can facilitate learning for slower and faster, more talented students alike. But the videodisc configuration goes one step further.

"Because the pictures are there on the computer, I remember the material better. It is easier to understand what is going on when you can see a picture of what is happening while it is being described. It's better than just being told about it because it's just as good as first hand experience. The demonstrations and procedures given on the videodisc were doubly good because of the visual demonstration of procedures and results. One of the exceptional characteristics of working with the laserdisc is its ability to generate superb graphics."

This gets right to the heart of what sets interfaced video CBI apart from other excellent computer lessons. The video is such a contrast from computer text and graphics that it seems to come right out of the screen. Nearly all shots are from the point of view of the student experimenter, and many are close-ups showing a pair of hands manipulating real glassware and familiar equipment. In fact, the proportions on the screen can give the illusion that the hands are those of the student or of a real live lab partner on the other side of the glass. At first glance, some students believed that they were being tricked by someone placing a hand behind the monitor and moving the chemicals.

"People can see what really happens - it allows them to make predictions and then see what happens. The experiment where I actually got to do it, sort of, let me get a grasp of what had just been demonstrated."

With this technology, students not only interact with computer text, but also with the video. Programs can be written to allow the video to be viewed in real time, fast forward, slow motion, freeze frame, or reverse mode at the whim of the observer. It can be initiated, terminated, or viewed at any time. These aspects can make it as valuable as, if not more than, hands-on experiments or demonstrations. Students do enjoy working in the laboratory, but much can be gained from close up viewing of an experiment with few distractions and the ability to correct errors at the touch of a button. Experiments can be performed repeatedly without waste of chemicals, setup or cleanup, and instructor fatigue. Several insightful students also suggested other applications.

"We can also use dangerous chemicals in experiments with no risk of being hurt...using equipment we don't have. The laser disc would be good for small or new schools who do not have a chemistry course."

Use of interactive videodisc in the chemistry curriculum presents many advantages and opportunities. Traditional CBI can be enhanced with the incorporation of live video, concepts can be better reinforced by linking them with real objects, and the versatility of the video allows observations impossible in a laboratory. Along with this, the instructor can use lessons to replace, supplement, or provide pre-lab exposure for experiments already in the curriculum. It is also quite easy to access a small segment of a lesson to use as a demonstration that again requires no set up or breakdown.

Currently, evaluation of videodisc programs in chemistry at this school has been limited to informal gathering of student reactions to lesson format and the machine in general and comparison of scores on pre and post quizzes. These preliminary tests indicate that learning does indeed take place and that students are very comfortable with the technology.

One student found "the interaction with the computer fun and provoking a good learning experience." Another "found it interesting, appealing and understandable." And a third that it was "entertaining as well as educational."

The next phase of evaluation will involve quantifying the amount of learning taking place, and testing what kinds of learning are possible. One such study would be to provide pre-lab exposure to a group of subjects whose performance in the experiment and on a quiz would be compared to a control group prepared by traditional means. Projects of this type have already been carried out at the introductory college level which show that presentation of course materials on videodisc can lead to achievement levels equal to or exceeding those of students in traditional classroom settings.[2]

---

1. General Chemistry
   Stanley G. Smith, Loretta L. Jones (producers)
   Compress (soon to be released)
   1-800-221-0419

2. Production and Evaluation of Interactive Videodisc Lessons in Laboratory Instruction
   Stanley G. Smith, Loretta L. Jones, Michael L. Waugh
   Journal of Computer-Based Instruction

GHOST WRITER

By: Robert Bortnick, CCSD #59, Arlington Heights, IL

Word processing is becoming as common an activity in schools as it has already become in offices. Word processing has demonstrated its usefulness in developing student's writing from the standpoint of increased interest in writing, greater creativity, and substantially more time spent on writing and rewriting activities. However, the teacher is the main ingredient in student's improvement in writing and that improvement is dependent on the frequency and quality of feedback that students receive on their writing.

For the past several years, District 59 has had the following four goals in its Language Arts Program at the junior high school level.

1. Spend more time on writing, less time on teaching grammar as an isolated skill.

2. Teach students to use computers in language arts as tools for writing.

3. Help students learn a strategy for writing (prewriting, writing, revision).

4. Train pupils to use computers to analyze paragraphs and essays to focus attention on developing writing skill proficiency.

To achieve these goals, one of three computer labs at each junior high is devoted to word processing. The labs are used most extensively in implementing the writing process including prewriting, writing, and revising activities. What remains is the need to provide the quality and frequency of feedback to students on their writing. To this end, a program was developed to provide students with aid in revising and editing their writing. The program designed for this purpose is Ghost Writer. Ghost Writer contains a set of utilities that provide the writer with feedback on a number of aspects of their writing related to its readability, acceptability, and clarity. Specifically, these utilities include:

1. Readability Test

2. Sentence Length Analyzer

3. Paragraph Outliner

4. Homonym Checker

5. Clarity Checker

6. Vocabulary Analyzer

The purpose of these utilities is to give students a tool to use in improving their writing. This tool provides them with the opportunity to identify many common errors and the ability to correct simple errors as part of the revision process. Such tools help students to assume responsibility for their writing and ultimately, become more independent.

Such a program also helps the teacher in several important ways. First, such programs free the teacher from the time consuming task of much of the basic editing of their students' writings (i.e., their for there). Secondly, teachers are freed to provide feedback on the logic, coherence, and unity of their students' writings. Research has demonstrated that teachers infrequently can or do provide this level of feedback.

As part of the implementation of the Ghost Writer program in the district's writing process efforts, data on students' writings has been accumulated on a longitudinal basis. Each student in the junior high during the 1985-86 school year wrote a composition under controlled conditions at the beginning and at the end of the year. These compositions were analyzed as part of the district's assessment program. During the 1986-87 school year, each student will again write a composition under controlled conditions at the end of the year.

A compilation of Ghost Writer analyses of students' writings is reported in the following two charts. These reports include analysis on Total Words, Readability, Vocabulary Index, Subordinating Conjunctions, Coordinating Conjunctions, and Maturity Index. The following definitions will assist in understanding the information reported in the charts.

Readability Test: Provides a grade level estimate of the reading difficulty of the writing. This estimate is based on the Fry Readability Formula and provides a grade level estimate of reading difficulty.

Vocabulary Analyzer: This utility gives the frequency of words used in the writing. The Vocabulary Index ranges from 0-100 and is derived from the ratio of number of different words compared to the total number of words in the passage. The higher the index, the greater the variety of words that are used to convey the meaning.

Clarity Checker: This utility provides a Maturity Index. The Maturity Index is an estimate of the level of maturity in the writing. This index is also a 0-100 scale. The higher the number of the index the better the writing is likely to be in terms of the increased frequency of subordinating to coordinating conjunctions.

The writing results in Chart 1 show the averages for students at sixth, seventh, and eighth grades in all three junior high schools. The mean scores under the various variables clearly indicate an increase consistent with expectations related to growing maturity in writing as students progress through the grades. In Chart 2 students at the seventh grade who have been identified by their teachers as above average, average, and below average writers are compared on several of the same variables. Again, the overall trends of the data are consistent with what would be expected for students in terms of different ability levels. Students with the least writing ability would be expected to write fewer words, have less diversity in their writing, and indicate less overall maturity.

In assessing the data, it was hoped that the vocabulary and maturity indices would provide meaningful assessment measures of students' writings. As George Hillocks points out, "Better measures to define and assess lexical growth in writing are needed. Many studies examined only syntactic aspects of writing ability in relation to reading achievement. None of the studies developed new measures of lexical maturity or complexity in students' writings." Based on the results of this study, there are two suppositions that follow with respect to the value of the vocabulary and maturity indices. The Vocabulary Index provides a sensitive index of writing with respect to the revision process. Throughout the year students were able to analyze their compositions, identify words used redundantly, and improve their vocabulary index score through revision of their early drafts. The limitation of the Vocabulary Index seems to be related to the length of the passage. The shorter passages show an inordinately high vocabulary index score as compared to what the student does in writing longer passages. The Maturity Index on the other hand did not reflect much change in the revision process. However, as seen in the data in Charts 1 and 2, the trends of these indices are consistent with overall age differences and ability levels. Students did show evidence of change in writing maturity level as measured by the Maturity Index over a period of a year with the greatest change occurring with the more able writers.

Therefore, the analysis of data using Ghost Writer seemed to provide some concrete means of assessing writing both in terms of the revision process and overall growth as a result of a year-long extended writing instruction. The data is also encouraging in supporting predicted trends in writing growth.

GHOST WRITER available through...

Robert Bortnick, Ph.D.
Assistant Superintendent for Instruction
School District 59
2123 S. Arlington Heights Rd.
Arlington Heights, IL 60005
Phone: (312) 593-4328

MECC
3490 Lexington Avenue North
St. Paul, MN 55126
Phone: (612) 481-3500
Ghost Writer
Catalog #A-153-037

GHOST WRITER

JUNIOR HIGH DATA

1985-86

GRADE	TOTAL WORDS	VOCAB. INDEX	READ-ABILITY	COORDI-NATING	SUBORDI-NATING	MATURITY INDEX
Sixth	62					
N =		60.8	4.0	5.7	2.9	29.6
Seventh	84					
N =		56.5	4.4	5.2	3.2	35.5
Eighth	92					
N =		58.1	5.4	6.1	4.2	37.8

CHART 1

GHOST WRITER

SEVENTH GRADE X ABILITY LEVEL DATA

1985-86

	VOCAB.		COORDI-NATING		SUBORDI-NATING		MATURITY	
	PRE	POST	PRE	POST	PRE	POST	PRE	POST
BELOW AVERAGE	56.8	54.8	4.2	4.9	2.8	2.8	33.5	36.4
AVERAGE	53.4	57.0	5.8	4.1	2.6	2.8	34.2	37.5
ABOVE AVERAGE	54.9	57.3	6.2	5.8	2.9	5.2	35.3	38.5

CHART 2

INSTRUCTIONAL GOALS FOR A HIGH LEVEL PROGRAMMING LANGUAGE

A POSITION PAPER

By: Don D. Bunt, Chicago State University, Chicago, IL

Recently I was asked to serve on an Illinois State Board of Education/Illinois Board of Higher Education joint task force charged with drafting goals in the area of mathematics for students entering baccalaureate degree programs. The goals were to be written in terms of student knowledges, understandings, and skills that resulted from pre-college schooling. Computing was a general area of concern for this task force. The idea for this paper was conceptualized while I was participating on this task force by formulating goals for any programming course taught in a higher level language.

I feel there is a great need to clarify what knowledges, understandings and skills students should possess after taking one or perhaps two higher level programming courses in the same language. Variability among instructional techniques should be encouraged so that instructors and students can be creative. But I am concerned that students who are not given sufficient direction often develop "bad" or "non-standard" techniques which "work", but which may interfere with future learning. "Bad" or "non-standard" techniques may also cause tremendous communication difficulties when the individuals participate in group projects while in school or later while employed as computer science professionals. Integration of logical design tools, structured programming techniques and appropriate use of documentation into any programming course taught in a higher level language is essential. Addition of these elements increases the complexity of classroom instruction, especially at the beginning of a course. However, it has been my experience that students who have mastered structured programming techniques complete programming assignments more independently, more quickly and are able to debug programs themselves. The payback for the careful control of course content is tremendous.

After compiling the list of goals for instruction of any programming course taught in a higher level language, I quickly realized that the goals could be rather naturally placed into seven categories: General Data Processing, Data Representation, Logic, Program Testing, Documentation and Files. These categories are not necessarily mutually exclusive nor are they exhaustive of all possible desirable goals. But they do represent what I feel are goals common to programming courses taught in a higher level language. It is obvious that these goals are not mandated, but they may serve as a starting point to motivate discussion regarding goals for instruction of programming courses taught in a higher level language.

Please notice that each objective begins with a behavioral verb. It should be possible to determine if students have attained these goals by observing their behaviors. The longest list of objectives occurs in the logic category, reflecting my philosophy that when good algorithms have been designed, modified and tested, the coding in the higher level language is not a major concern.

Students should be able to:

I. **General Data Processing**

1.1 Understand that a computer program communicates with its environment.

1.2 Describe several Input/Output devices.

1.3 Understand that a compiler translates source code written in a higher level language into machine language for a specific machine.

1.4 Know whether listed errors were written by the compiler or the operating system.

II. **Data Representation**

2.1 Understand and use appropriate standard data types: INTEGER, REAL (floating point and exponential), CHARACTER, CHARACTER STRING and BOOLEAN in designing solutions to computer programming problems.

2.2 Implement an algorithm using records in a higher level language.

2.3 Implement an algorithm using a single dimensional array in a higher level language.

III. **Logic**

3.1 Understand that program logic is machine and language independent.

3.2 Design algorithms as a sequence of the standard logical structures: LINEAR PROCESSING, IF-THEN-ELSE, CASE and DOWHILE.

3.3 Use flowcharts or pseudocode to design and document a solution to a computer programming problem.

3.4 Break complex algorithms into functional logical modules.

3.5 Understand and use Boolean expressions in the solution of a computer programming problem.

3.6 Understand and use the logical connectives "OR" and "AND" and the negation operator "NOT" in writing compound statements.

3.7 Use a structured top down approach when designing a solution to a computer programming problem.

3.8 Understand and use the order of operations when evaluating Boolean expressions.

3.9 Test the adequacy of the logical design of each program module by "walking through" the logic a step at a time prior to coding in a higher level language.

IV. **Program Testing**

4.1 Design sets of input data which test for a valid program.

4.2 Compare program output with program specifications listed in the programming problem.

4.3 Verify computational results listed in program output with a hand held calculator.

V. **Structured Programming**

5.1 Understand and use the fact that each IF-THEN-ELSE, CASE and DOWHILE logical structure, and also each subroutine, has exactly one exit point.

5.2 Code structured algorithms in a higher level language.

5.3 Code the IF-THEN-ELSE, CASE and DOWHILE logical structures in a higher level language.

5.4 Implement subroutines in a higher level language.

5.5 Debug compiler and run time errors.

VI. **Documentation**

6.1 Read and use software and hardware documentation presented
a) in vendors' manuals, textbooks and user's guides and
b) interactively by running computer software.

6.2 Use meaningful names for variables in algorithms and higher level language code.

6.3 Describe input and output files and parameters, and the purpose of each program module including the main routine.

6.4 Write comments in the higher level language code to explain the purpose of each non-obvious program entity.

VII. **Files**

7.1 Know and use the structure of an external disk text file.

7.2 Format program text output with appropriate centering, spacing, labels and headers.

7.3 Know and use the structure of an external disk file of records.

THE EFFECT OF WORD PROCESSING ON THE ACQUISITION OF COMPOSITION SKILLS BY ELEMENTARY STUDENTS

By: Maureen Cheever, Northbrook SD #28, Northbrook, IL

Does the use of a word processor improve the quantity and quality of the composition skills of fourth graders more than the use of the more traditional means, pencil and paper?

Does the use of a word processing program increase the number of revisions? What type of revisions are made? Is an increase in revisions related to an increase in the quality of writing?

Will students be able to transfer the experience of revision, pre-writing, and test analysis to pencil and paper writing? Should this skill transfer be a goal of computerized composition, or is it unimportant?

Do students' attitudes toward writing vary with the process used (word processing versus pencil and paper)?

What are the teachers' perceptions about the place of word processing in the composition curriculum?

Is there anything unique about the environment in which word processing takes place?

Should word processing be included in the elementary curriculum?

ABSTRACT

These were basically the research questions involved in the author's doctoral study. The purpose of the study was to examine the effects of using a word processor on the acquisition of composition skills in seven classes of fourth grade students. Three classes used a word processing program, MAGIC SLATE, for at least one 40-minute session per week for the duration of the school year, while the other four classes did not use the word processor in their composition curriculum. Pre-test and post-test pencil and paper writing samples were collected from all students, as well as post-test word processed samples from the experimental group. Randomly selected samples were scored for quantity and quality in four ways (length of text, holistic rating, conventionality errors, and spelling errors), and analyzed by ability. Observations were made to identify the number and type of revisions each group made while writing. Students' attitudes toward writing and teachers' perceptions of word processing were assessed through separate surveys.

Significant findings occurred in the experimental group in all measures for the quantity and quality of writing. The most significant finding was a 70.4% reduction in conventionality errors and a 60.2% reduction in spelling errors in the experimental group's pre-test/post-test scores. The greatest gains in reduced spelling and conventionality errors were demonstrated by students with the lowest ability scores. Attributes of students who wrote longer and higher quality texts include a preference to use the computer for writing and an attitude that the act of typing does not interfere with their writing.

New to the body of research are the findings which document the teacher's view of the use of word processors in the elementary curriculum. Most significant was that the teachers felt that word processing should be a part of the elementary composition curriculum.

Finally, addressed briefly in this research was the environment of the computer lab. It took some adjusting by the teachers to NOT view the lab as being similar to their regular classroom. As the "revision observer" noted: "The teachers kept trying to enforce a code of silence in the lab. However, the students consistently broke through the attempts in order to collaborate about their writing with fellow students." The author observed sessions in the lab which were active and noisy and others which were intensely quiet. Teachers will want to manage the environment to best support each individual lesson.

IMPLICATIONS

Several significant findings occurred after the pilot year of integrating computers and composition. The author feels strongly that these findings are just scratching the surface of the potential that exists. Educators have seen this potential in their own classrooms, yet previous research failed to document this contention. However, the short duration period of most of the studies tainted those findings. This study lengthened the duration of the experiment to one school year and significant differences were found. Now the task is for teachers, administrators, and researchers to devise strategies to best support and utilize word processing in the elementary classroom.

SUPPORT

Support for integrating word processing into the elementary composition curriculum involves the participation of administrators and teachers together. First, access to the computers and sufficient time on task must be organized. Piper's study (1984) noted that having four computers and printers for each language arts classroom left much to be desired. Prior to the organization of the pilot lab in this study's cooperating district, teachers complained of inadequate support for group instruction and participation with just a few computers in their classrooms. In this study, eleven computers were available in the lab and each grade level had at least one computer for use in their individual classrooms. Additionally, access to the computers includes organizing sufficient time-on-task. In this study, the teachers had one reserved 40-minute session per week, and, in addition, they generally signed up for at least one more session each week.

Second, the support of the building principal is fundamental for the program to take roots. In the author's experience, the greatest success in integrating computers into the composition curriculum occurred in the school whose principal exhibited an interest in the program. Frankly, this does not mean that the building principal need be a "computer expert". The principal should be involved in general decisions about using computers in the curriculum and, most importantly, should support any efforts toward implementation.

Third, a computer coordinator or consultant whose main responsibility is to educate and support the teachers is critical. This person should meet with each teacher, determine what the individual's computer needs are, and design a plan to meet those needs. Follow-up should be scheduled on a regular basis.

UTILIZATION

Strategies to best utilize the power of word processing include the teaching of touch typing skills, teaching revision skills as a part of the writing process, and modifying former techniques of writing instruction for optimal implementation.

The teaching of proper touch-typing techniques will be essential for students to become fully competent as word processors. Although student attitudes and research indicate that students can become successful word processors without knowledge of these skills, the

author believes that in years to come, NOT being proficient in keyboarding will be a hindrance for student writers. These skills can be introduced and reinforced through software programs currently available on the market. The cooperating district for this study has implemented touch-typing lessons beginning at the third grade level, expecting mastery by the fifth grade.

Second, specific revision skills must be taught if students are to be expected to use them. In this study, it became clear that the teachers assumed the students knew how to proofread. However, students had to be shown over and over again that revision meant more than finding spelling errors. Perhaps these are lessons which could be introduced in the classroom prior to going to the lab. In that way, when the teacher gets to the computer lab, she/he can spend time enhancing the student's writing product instead of being "caught up" in meaningless revisions.

Finally, through this research, it became apparent that the teacher must consider some subtle differences when teaching writing via the computer.

The first item is an awareness that the child's writing is open for public scrutiny due to the computer monitor. Unlike a sheet of paper that a student can shield with a body block, writing on the computer is difficult to keep as a private thing. As the author read the writing samples for this research, the responses sounded very similar. One would hate to think that "socialized" writing was to be a by-product of word processing.

Additionally, through the student attitude survey, it was found that students do not always like to write in pairs. Thus, teachers should not press for collaboration in every writing activity.

And last, a reminder from Boone (1985): "The word processor does not set the atmosphere for writing. It does not give encouragement and praise. These remain in the realm of the teacher. What the word processor does is provide a writing environment in which children can explore and play with their language."

FURTHER RESEARCH

Suggestions for future research in this area include: (1) If students are to respond to a "trigger" question, develop one that is very open-ended and can touch the students' emotions--the robot question was too simplistic ("If you could invent a super robot, what would you make it do?"); (2) The author strongly suggests a similar study organized with the first or second graders over the course of three or four years; and (3) The author recommends that research similar to this use a text-analysis program--one that would assess readability, length analyzer, vocabulary analyzer, for example, GHOSTWRITER (Bortnick, 1986)--an an evaluative measure.

In closing, Becker (1984) has stated: "The development of computer-based writing tools may lead to a better understanding of how to improve student motivation to write, writing habits, and writing styles. This knowledge might lead to other programs that implement these principles without the costs associated with technologically sophisticated means. In other words, computer-based writing tools may be a means not only for students to become literate, but also to use literacy to make a productive contribution to their surroundings." The author firmly believes that this research has captured evidence at just the beginning of a tremendous surge in educational technology. Through word processing, elementary children have the opportunity to achieve and excel in written communication.

REFERENCES

Becker, H. J. (1984, March). School uses of microcomputers. Educational Technology, pp. 33-34.

Boone, R. A. (1985). The revision processes of elementary school students who write using a word processing computer program. (Doctoral dissertation, University of Oregon, 1985). Dissertation Abstracts International, 47.

Bortnick, R. (1985). Ghostwriter [computer program]. St. Paul, MN: MECC.

California Test of Basic Skills, Form U (1981). Monterey, CA: McGraw-Hill.

Educational Testing Service (1976). How the essay in the CEEB English composition test is scored. Evanston, IL.

Piper, K. L. (1984). An investigation of the effects of sentence combining and story expansion delivered by word processing microcomputers on the writing ability of sixth graders. (Doctoral dissertation, Texas Tech University, 1984). Dissertation Abstracts International, 45, 2747A.

Sunburst (1984). Magic Slate. [computer program].

INTERACTIVE TOOLS--FOR A SCIENCE THAT COMES ALIVE

By: Ann S. Dana, Elementary School District #181, Hinsdale, IL

Thermistors, Photo Cells, Galvanic Skin Resistance Devices, Seismoscopes, on screen Tachometers and Speedometers, interface the computer to science processes. Students use the computer to collect relevant data that can then be analyzed. By using Broderbund's Science Toolkit and the additional modules, plus programs by DCHeath and Sunburst, science experiments become exciting. The computer can store, calculate, manipulate and print out data. Students can then question why? Experiments can be replicated easily. There is more process going on then working by rote.

Being able to use the microcomputer as a laboratory instrument in science is an important use of the microcomputer in education. With the use of the MICRO/scope box or the Interface Box (depending on which software package) students have the opportunity to generate real-time data that will be graphed on the screen and can be printed.

The general purpose of these software packages is to help students see the computer as a tool they can use. Today, most scientists, engineers, and medical technicians use computers to analyze data, store records, and solve problems. Computers are readily available today and this speeds up data used in diagnosing, monitoring, designing, and testing materials. On the other hand, doing experiments and recording the data by hand is time consuming and tiresome. The computer can relieve these hardships.

The interface provides access between the computer, the physical world, and the student. By connecting sensors, the computer can analyze and display data. The student can control the sensors and formulate problems to work on.

The computer can be adapted to simulate the expensive equipment being used in real research laboratories. This introduction to scientific research can help students to realize that there are problems that must be recognized and now there is a means of designing experiments that may help to solve some of them.

The interfaces in the science units are the results of developments at TERC, Technical Education Research Centers, in Cambridge, Massachusetts, headed by Bob Tinker.

There are several commercial packages available. One is Broderbund's Science Toolkit. It consists of a diskette, a manual with several suggested experiments, and 2 probes, a thermistor to measure heat and a photocell to measure light. The probes are plugged into a box that interfaces with the game port of an Apple II+ or IIe. Temperatures of various solutions can be charted or graphed. Think of all the science units where temperature is a factor. The students calibrate the thermistor and then use the instrument to automatically read the temperatures. They can be recorded at set intervals for an extended period, even 24 hours. The graphs can be rolled back. The points can all be printed or certain portions can be selected to save paper. The line graph appears as students connect the points printed. An additional Teacher's Guide is available for $30.00. It gives more information on conducting the experiments that are described in the Experiment Guide that comes with the Science Lab. The lab retails for $59.95.

Two additional kits are available from Broderbund. They use the equipment from the Science Toolkit. Module: 1 Speed and Motion and Module: 2 Earthquake Lab each use the photocell. There are an 'assemble it yourself' seismoscope and a speedometer/photocell stand to put together, as well as the experimental car from the Speed and Motion Module. Disks accompany each module manuals of experiments. Prices are $39.95 per kit. It is necessary to have a light source and a fan to conduct experiments with these modules.

Exploring Heat by DCHeath correlates with Heath's fifth grade Science or Chapter 12 of their Physical Science (7th-8th grades). However, there is a complete Heat Unit with 10 teaching lessons in the manual. The Dictionary program on the disk is a tutorial in temperature and heat. There are 3 programs that make use of the 2 thermistor probes that accompany the software and manual. Molecule Window shows what happens to matter as the temperature varies from freezing to steam. Thermometers and Temperature Sketch graph the readings in bar or line graphs which can be printed. One unique feature is that the teacher can create individualized disks for students. Initially the student needs to boot the Teacher's Disk but then can use his own for the programs he needs. The probes must be calibrated and then used only at that computer or recalibrate. Ideally it would be best to have enough units so there are no more than 2 or 3 students at a computer. That would make it very expensive! The program is $150.00, but it is worth the investment to have such a dynamic way to have students discover how heat and temperature are related.

A third software package that uses thermistors is probably the most reliable or most closely resembles what scientists would experience. This is HRM's, Human Relations Media, Experiments in Science. The one used was a prototype of what today is more sophisticated. This 1983 version has 12 experiments of varying difficulty. There are three experiments in each of the 4 science areas of chemistry, physics, earth science, and biology. Each experiment includes objectives, description, procedure, questions, and extensions. The equipment includes 2 thermistors and 2 light sensor strips. The package is $259.00. HRM has several very sophisticated units in biology, chemistry, and physics.

A biofeedback program that interfaces to the computer is Sunburst's Learning to Cope with Pressure. It comes with a Galvanic Skin Resistance Biosensor. Stress is a syndrome that concerns everyone. This program will let us help students to recognize causes of their anxiety and, therefore, learn to relax and reduce stress. The software includes a Physical Stress Test, including a Muscle Stress Test and Reaction Time Test; Psychological Stress Test, includes 2 word association tests; and Stress Managements, teaching how to relax through CalmPatterns, CalmScope, and CalmBar. As the programs are used, the GSR helps to record the changes in stress read through the skin. The graphs can be printed out. The manual gives ideas on how to use the program in your classroom, helping your students to learn to relax. The program with the GSR is $99.

At the junior high level we did a comparison study on the three software packages that contained thermistors. The students were to discover what would be the best insulation for a glass container. They were divided into three groups, one for each software package. Each student insulated a glass. Materials available were polyurethane foam, feathers, styrofoam kernels, newspaper, foil, plastic wrap, dark paper, wool, and other fabric. There were lids for the containers. Each group calibrated their probes, measured the temperature of an uninsulated glass of water for 10 minutes and then tested their insulated containers. The students wrote up their results using AppleWorks, discussing problems, how well the group worked together, and conclusions. Temperature results were put into a student constructed data base. One of the conclusions was that foam made the best insulation. HRM and DCHeath programs let students calibrate with boiling water and ice. Broderbund's Science Toolkit does not allow students to use the thermistors in extreme temperatures. They feel it is not safe for the students. The students final project was to

write a fictitious account of their being lost in either a northern latitude or a tropic zone. They were to take into account what they learned about insulation and how this information helped them in their adventure.

Apple Education Affairs has produced a resource guide, Microcomputer-Based Laboratories, written by Robert Tinker of TERC. It contains excellent descriptions of a MBL Lab, experiments, software, sources of supplies, and 'The Educational Issues'. The booklet contains the BASIC programs and the instructions for building and using a light sensor. You can build one and conduct your own experiments in light measurements. Write to Apple at 20525 Mariani Avenue, Cupertino, California, 95014.

Learning science by 'doing' is a value that cannot be replaced. When students actually visualize the effects of what they do, they gain a better sense of what change means and can then learn to predict. The graphic pictures of their experiments will help them gain the significance of what graphs mean and how to use them to understand relationships.

SCIENCE TOOLKIT (Broderbund Software)
17 Paul Drive
San Rafael, CA 94903-2101
415/479-1185
$59.95

EXPERIMENTS IN SCIENCE (HRM Software)
175 Tompkins Ave.
Pleasantville, NY 10570
1-800-431-2050
$259.00

EXPLORING HEAT (DC Heath Software)
125 Spring St.
Lexington, MA 02173
617/862-6650
$150.00

HOW TO COPE WITH PRESSURE (Sunburst Communications)
39 Washington Ave.
Pleasantville, NY 10570
914/769-5030
$99.00

SEARCH AND RESEARCH - - IS A DATA BASE A VIABLE PART OF THE ELEMENTARY CURRICULUM?

By: Ann Dana, Elementary School District #181, Hinsdale, IL

Have you tried to find a particular book by using a library's 'touch sensitive screen' access file? By the warmth of your fingertips, you can quickly find the author, the book, and an abstract. It isn't that we're in such a hurry, but it's just an easier way to find exactly what we want out of the wealth of information available to us today.

Data bases have been with us for a long time. Every time you sort photographs by years or checks by numbers, you've created files. The electronic data base is needed when there's such an abundance of information that it no longer fits into that shoe box. It's a way to help us organize, classify, up-date information. It makes it easier to access, to look for trends, to discover likes and unlikes, to help us question, to weed out the unnecessary, to look for patterns, to draw conclusions.

Using a data base in the classroom can help students take part in activities that relate to the adult world. Another very important step in using the data base is as a tool to help students think critically.

Beverly Hunter, President of Targeted Learning Corporation, has been an advocate of the use of data bases and critical thinking even before the microcomputer revolution began. Her book, My Students Use Computers, Reston 1984, is but one of the resources she has written with the importance of data bases as part of the content. She was a consultant for the Technology in the Curriculum manuals developed in California in 1986. She helped author the "Immigration" article in Teaching and Computers's January issue. She has a newsletter of worth called DBKids. It's a way to have each of us share what's exciting with data bases in our classes.

The 5 volumes, titled Technology in the Curriculum, were published and distributed in 1986 to all California K-12 public schools. One manual and diskette, the **Data Relator,** is a data base that can be copied and used in all subject areas. The disk contains a data base covering all materials in the manuals. You can search by a particular subject. The curricular areas covered are History-Social Science, Science, Language Arts and Mathematics. In the History-Social Science volume, as in all volumes, objectives for study, interpersonal social participation, critical thinking, and content skills, plus software and Instructional TV programs that can be integrated with other instructional techniques to enhance student learning, are completely described. Sample lesson plans are also included. The plan for Grade 5 incorporates **PFS: File** and **PFS: Report** to be used as students gather data about famous Americans. By using the ITV Series, "Truly American", the class will define their fields and enter the information for one of the characters. Each student will then research their assigned famous American and enter it into the data base. They will formulate questions for a search. Suggestions are given for Independent Practice and a Closure Activity. The **PFS Data Base U.S. History** by Scholastic is in the Grade 11 Lesson Plan.

Commercially there are several choices of data base software. Grolier's, **Friendly Filer**, is a good tutorial, easy to use, but limited. It has additional templates that can be used with the filer.

Appleworks by Apple is easy to set up and use. You are limited in the length of data you enter per field. For younger students, who may be more chatty, **PFS: File** gives you unlimited space for data entries. It, however, isn't as easy to set up and a disk can only hold one file. You also have to have **PFS: Report** if you want to print a report in columns. Scholastic has developed some excellent data base data disks to accompany **PFS: File**. They hold extensive information. **U.S. Government** has 5 separate data base activities: Federal Spending, Presidential Elections, Constitutional Convention, State and Local Governments. The Animal Disk can be used with most grade levels by setting the guidelines. The information it contains is very extensive.

The newest data base is Sunburst Publications', **Bank Street School Filer**. They have taken Bank Street Filer and added a whole new teaching dimension. There are three versions, beginners, 64K and 128K. The beginners is for those that do not want to bother with defining fields, producing a variety of reports, or adding comments to the records. The 64K and 128K are similar except the 128K must be used with an Apple or Commodore that has that much memory. There are some real pluses with The School Filer. You can define your fields. Number, money, date, will accept just those. Data entries can fill the 80 columns. A comment field can be added that will hold 255 characters. You can move your fields to new locations, rename fields without losing data, merge records easily from one disk to another. You can compare up to four fields for reports. By pressing ESC from the main menu, you can browse between records. The data disk can hold several files. Its menu is similar to **Bank Street Writer**. Data base reports can be merged with **Bank Street Writer III** word processing files and also with 80 Column **Magic Slate**.

Using data files that have already been created by someone else is one of the best ways to introduce students to data bases. They can sort, search, test hypotheses, discover relationships, help solve problems, investigate new subjects and much more. By using one of Sunburst's data disk packages with **The School Filer**, you and your students can begin to realize the power of a data base. Try the **United States Databases**. Each record has 20 fields of information. Let them browse, find their own state. Prepare some questions for the students to research:

In what state is the Ohio River? How many states are in the SE Region?	Knowledge Question
If you wanted to build a ski resort, where would be the best location?	Evaluation
Compare 3 states by their chief, products, natural resources, and populations. What relationships do these categories have?	Analysis
Show how cities in other states are similar to your state's largest city	Application

The second step in using a data base is building one. A file can be as simple as a class personnel file. Fields could be:

NAME
PHONE
BIRTHDAY
STAY FOR LUNCH
BUS RT.#
LOCKER
FAVORITE FREE TIME ACTIVITY

Students could input their own information and it could be easily accessed for up to date monthly birthday file or bus route slips. Excellent source ideas can be found monthly in Scholastic's Teaching and Computers. Some examples:

Feb. '85 - Dinosaur	May/June '86 - Country, Travel
April '85 - Stars	Oct. '86 - Pond Life
Nov. '85 - Indians	Jan. '87 - Immigration
Feb. '86 - Presidents	Feb. '87 - Black Americans

A third step will be when students can design their own files to meet their individual needs. It could be a data base file that correlates with research they are doing for a report. It might be just a bibliography of materials used in the research. Questions have to be formulated first to determine what fields will be needed. It is necessary to be as specific as possible, to be clear about the information that will be collected. An article by Soloman, Electronic Learning (1986) suggests librarians can help students create a data base when doing research. The fields might be:

TYPE OF INFORMATION
ARRANGEMENT
SPECIAL FEATURES
LOCATION

In using **Bank Street School Filer**, data bases can be created by students, teachers, or used with already developed templates, to match most curricular areas from grades 2 up. By using higher level questions, children search their data bases and apply this knowledge in extended activities. The variety of knowledge gained widens their horizons and makes them more aware of how information can be obtained.

SOFTWARE INFORMATION

APPLEWORKS $250.00
Apple 7 - 12
20525 Mariani Ave.
Cupertino, CA 95014
1983 Data Base, Word Processor, Spreadsheet (Integrated)
Integrated, user can move easily from one to other, can merge reports into word processing document. Disks can contain several files, which can be merged. Label or table reports. Holds up to 1,350 records/file. Sorting, selecting, finding, comparing (on 3 fields), contrasting.

BANK STREET SCHOOL FILER $99.00
Sunburst Publications (lab pack of 10 $297.00)
39 Washington Avenue 5 - 12
Pleasantville, NY 10570-9971
1986 Data Base
Includes tutorial, extensive manual, data disks of Classroom Tools, School Data. Easy to assign type of field. Can add fields, rename fields. Browse with ESC. Merge reports to Bank Street Writer, Magic Slate. Sorting, comparing and contrasting by 4 fields, reports, evaluating.

FRIENDLY FILER $54.95
Grolier (lab pack of 5 $109.95)
Sherman Turnpike 4 - 12
Dept. 333
Danbury, CT 06816
1984 Data Base
One disk data base with interactive tutorial with game for students to use to sort, select, and search. Good to use before trying more flexible data bases. Only holds 7 fields, 350 records per file. Sorting, searching, contrasting, comparing.

PFS: FILE $99.95
Scholastic, Inc. (lab pack of 5 $185)
730 N. Broadway 5 - 12
New York City, NY 10003
1984 Data Base
Create form, then fill in data. Extensive teaching manual. Holds up to 1000 records per disk. Only one file per disk. Can be used with PFS programs, Graph, Write, Report, Access, Proof, Plan. Designing, analyzing, sorting, comparing, contrasting.

TECHNOLOGY IN THE CURRICULUM $95.00
California State Department of Education K - 12
P.O. Box 271
Sacramento, CA 95802-0271
1986 Tool, Teacher Reference Guide
5 volumes, one a data base, covering software, ITV within curriculum. Volumes cover History-Social Science, Language Arts, Mathematics, Science. Complete descriptions plus sample lesson plans. Sort, classify, search with data base.

RESEARCH VS. PRACTICE IN EDUCATIONAL COMPUTING
A GAME WITH NO REFEREES

By: Allen Davis, Lake County Educational Service Center, Grayslake, IL

Educational computing has developed in a hap-hazard manner. Most current programs had their origins several years ago when one faculty member with a keen interest in computers would convince the administration to purchase a microcomputer. It was not unusual for that computer to be in the mathematics arena with BASIC programming the chief activity. In those days, microcomputers were more a hobbyists' device than an educational tool Because the initial interest in computers was more a 'bottom up' movement, there was little coordination of computer activities among educators. The goals and objectives were vague at best and commonly non-existent. The curriculum was under constant change as the availability of computers, the development of software, and the pressure of parents increased. As school boards were confronted with unprecedented costs, administrators started looking at goals, curriculum, staff training, and evaluation.

In August of 1986, the "Educational Technology Task Force Report" of the National Governors' Association stated the following goals (among others) for 1991: School districts will develop plans for applying technology and the plans will--

1. Identify what performance level these schools intend to achieve, how technology can assist in these achievements, and its integration into the ongoing curriculum development.

2. Provide for teacher training during times convenient to all teachers.

The goals imply that districts will need to develop a written plan on how they will use a variety of technologies, the objectives to be achieved, an evaluation process to determine the impact of the technologies, and the various staff development activities needed for staff training.

The introduction and use of microcomputers in educational settings has been a unique experience for schools. The technology is changing so rapidly: How much memory is enough? How much better graphics are necessary? What is the best sound quality needed? Are more printers needed? When do we stop purchasing computers? This list of questions is nearly endless, but on the minds of board members and administrators.

Some say that today's students live in an increasingly rich computer environment; that our students are surrounded by computers in their everyday life. Thus, the curriculum will change because of the computer background of the students. This will force educators to adjust to the changing curriculum, new instructional approaches, and computer-sophisticated students. Other concerned educators are not convinced that schools will automatically change to meet these new challenges.

One practical drawback in educational computing is that schools are dependent on outside forces to develop the technology, i.e. private sector manufacturers produce the computers. There is no 'school computer,' designed under specifications set by educators. Thus, no standard keyboard, monitor, operating system, etc. exists. It may be that different computers are needed in different grades and in different settings, depending on the objectives of the curriculum.

Currently, there is a great debate as to the purpose of computers in elementary and junior high schools. One group believes that computers should be used for exploration, and that programming, as a vehicle for problem solving, should be emphasized. Most of these advocates feel that children should have time to discover through experimentation, not burdened with course objectives, time-lines, and scope and sequence charts. Another group feels that computers should be used as a tool--whether applications, CAI, or other software dependent modes. There is little research supporting either view. As school curriculum personnel attempt to design a curriculum, this lack of research can cause serious problems. As boards and administrators ask for evidence, little is actually available to the degree that persons would put their job on the line based on the research.

Those in the field have observed the following in recent years: The teaching of computer programming is decreasing nationwide. The dominance of the mathematics faculty in using computers is decreasing. The use of computers in English classes is increasing. The use of computers in both art and music is just beginning to increase and will continue to increase. More and more, word processing and data base applications are becoming the norm. Some are amazed that little of these curriculum thrusts have been researched. More people probably should be concerned about the lack of research before even more changes in the curriculum should occur.

What makes a successful education computing curriculum program? The following eight factors seem to be present in many schools with successful programs. Not all factors necessarily need to be present, but the more that are, the more likely a successful program exists. There are certainly other factors that play an important role in a successful program, but these eight seem to be the most critical.

1. The district has a board-approved 3-5 year plan which includes goals, curriculum, hardware, software, training, personnel, maintenance, and evaluation. The plan is updated each year.

2. The visible support of the building principal.

3. A full time district computer coordinator.

4. A strong, active and flexible staff development program offering inservice to staff.

5. Computers available to staff in offices, in the lounge, to take home, and in classrooms.

6. Computers available for instruction in a lab, on carts, and in the learning center. Other hardware such as printers also available.

7. Legal software fitting the curriculum readily available.

8. A written curriculum stating where and when computers should be used in all appropriate classes.

As one investigates the following current uses of microcomputers, one should keep in mind the eight factors. Today, many schools are using computers as follows: keyboarding instruction, word processing instruction and applications, writing assistance, LOGO, problem solving software, CAI in general. If one looks at experimental research, CAI is the best researched and LOGO has also been researched. There is rather clear evidence that CAI does have a significant effect on learning in some settings. Hard data showing that the transfer of thinking skills learned in LOGO experiences to other situations is not at all

convincing. A warning--if a board of education requires strong research to justify a program, such evidence is probably not strong enough to satisfy a critical board.

Look at the eight factors again as some of the 'cutting-edge' topics are listed below: networking, telecommunications, site licensing, and large screen projectors.

This wonderful technology is available. We want to have students become familiar with it, to use it, and to control it. But what about another vital part of education? Do we have research on appropriate teaching methods and instructional strategies in teaching so many areas of educational computing? Appleworks is taught in graduate school and in 3rd grade. Where is the appropriate place for such a topic? Can curriculum specialists prove where the best place is?

As one investigates research projects in the field of educational computing, a question to be asked is: Who is doing the research? Is it being conducted by universities and research institutions, software developers, hardware manufacturers, magazines, or local school districts? The design of the research is very important. Is it survey research, experimental design, or meta-analysis? Are there biases, lack of control, vague hypotheses, or novelty effects? There is a lot of investigating going on. Many dissertations at universities are unpublished and difficult to obtain. The National Diffusion Network attempts to inform schools of significant programs. The Journal of Educational Computing Research attempted to provide a source of research reports. The Computing Teacher has started a section on research.

The final outcome is this--there is a vast amount of computer usage in practice, but little of it is truly research based. We live in an affluent, but practical society. We can try new ideas and afford the cost. We want to provide our students with the latest and most up-to-date equipment we can acquire. But if questioned closely, educators would have a difficult time justifying from a research basis much of the equipment, time, and personnel used in educational computing.

GRAPHING MADE EASY

By: Walter R. Dodge, New Trier High School, Winnetka, IL

An IBM PC laboratory with 12 microcomputers was used to provide participants with hands-on experience using a sophisticated two-dimensional graphing program written by the speaker. A brief introduction on the various capabilities of the program were provided and several samples to show the capabilities were explored.

Participants were then shown several unique applications where the program would be valuable within the high school mathematics curriculum. These applications were then performed using the computer by the sessions' participants.

Finally a series of problems relating to the mathematics curriculum were given to the participants to explore on their own using the graphing program.

The following pages include the introductory materials and the worksheets that the participants used.

INTRODUCTION

The Omnifarious Plotter is a powerful two-dimensional graphics program which will quickly graph most equations encountered in grades 7-14. This program can graph the following types of equations:

$y = F(x)$
$x = F(y)$
Conic Equations
Polar Equations
Parametric Equations

The program includes many features that make it valuable both for students and faculty. The program is entirely menu driven for ease of use. It is easy to learn to use yet powerful enough to handle most all of your graphing needs. Up to 10 graphs may be drawn on any one screen and it is possible to mix any of the types of equations mentioned above on a single set of axes.Color is used throughout to enhance the graphics image. In addition, axes values are easily changed and for polars and parametrics the values of the parameters θ and t are easily adjusted.

Equation entry is in normal mathematical form, so no computerese is necessary for equation entry. Exponents appear as normal exponents, functions appear in normal mathematical form, the constants π and e are entered as symbols, and multiplication is understood by adjacency so no "*" notation is necessary.

Graphs may be saved and retrieved from disk and also may be printed out to a printer.

It is possible to trace along an equation that has been graphed and obtain an ordered pair readout of points along the curve. Using this feature you can find approximate values for zeroes and extrema of a function. When graphing polar or parametric equations you also are given values of the angle or values of the parameter t for points along the curve.

The program also includes a Zoom feature so that you can move in for a closer look at a section of a graph or graph(s). This also, in conjunction with the Trace feature, allows you to obtain more accurate values for the points of intersection of graphs.

OMNIFARIOUS PLOTTER

OPERATIONS

-	Unary Minus	
-	Subtraction	
+	Addition	
*	Multiplication	(Also may omit this and the computer will understand multiplication by juxtaposition)
/	Division	

Shift 6 **Rational integer exponents**. Use this type of exponent for integer or rational constant exponents written in a/b form.

Alt 6 **Real exponents.** Use this type of exponent when entering decimal value exponents or functional exponents.

FUNCTIONS SUPPORTED

(1)	Sqrt (x) or √ (x)	For finding square roots of an argument.
(2)	Exp(x) or x e	For finding e to an argument.
(3)	Abs(x) or |x|	For the absolute value of an argument.
(4)	Int(x) or [x]	For the greatest integer of an argument.
(5)	Sin(x)	For the Sine of an argument.
(6)	Cos(x)	For the Cosine of an argument.
(7)	Tan(x)	For the Tangent of an argument.
(8)	Cot(x)	For the Cotangent of an argument.
(9)	Sec(x)	For the Secant of an argument.
(10)	Csc(x)	For the Cosecant of an argument.
(11)	ArcSin(x)	For the ArcSine of an argument.

(12)	ArcCos(x)	For the ArcCosine of an argument.
(13)	ArcTan(x)	For the ArcTangent of an argument.
(14)	ln(x)	For the natural logarithm of an argument.
(15)	Log(x)	For the base 10 logarithm of an argument.

SPECIAL KEYSTROKES

(1)	Alt T	For the greek letter θ .
(2)`	Alt S	For the square root symbol $\sqrt{}$.
(3)	Alt A or Shift \	For the Absolute Value bar.
(4)	Alt P	For the mathematical constant π .
(5)	Alt E	For the mathematical constant e.
(6)	Shift 6	To begin a rational integer exponent.
(7)	Alt 6	To begin a real or functional exponent.
(8)	DownArrow	To end either type of Exponent. This is found on the numeric keypad at the 2.
(9)	BackSpace	To correct an error on an input.
(10)	Esc	To back out of a menu when done or when entering a menu by accident.

WORKSHEET

Problem #1

(a) Graph: $y = 5\text{Sin}(x)$

Graph: $y = 3\text{Cos}(x)$

(b) Change axes values

$x \in [-2\pi, 2\pi]$
$y \in [-6, 6]$
x Tic $\pi/4$
y Tic 1

(c) Regraph all equations

(d) Trace

(e) Zoom Factor - 2
About point of intersection in first quadrant

(f) Reset all values to defaults

PROBLEM #2

Solve for all values of x.

$$|x - 3| - 2 = -2|x + 1| + 8$$

(a) Graph: $y = |x - 3| - 2$

Graph: $y = -2|x + 1| + 8$

(b) Use trace to get approximate x values for solution

(c) For more accuracy use Zoom and Trace

(d) Clear and Reset

PROBLEM #3

Show that $\sqrt{(x^2)} = |x|$

(a) Graph: $y = \sqrt{(x^2)}$

(b) Graph: $y = |x|$

(c) Clear and Reset

PROBLEM #4

(a) Graph: $y = 1/2\ (x - 2)\ (x + 4)$

(b) Clear and Reset

PROBLEM #5

(a) Graph: $y = 4\ln(x - e)$

Remember the constant e is gotten by Alt e.

(b) Clear and Reset

PROBLEM #6

Solve $x + e^x = 10$ For x.

(a) Graph: $y = x + e^x - 10$

(b) Use Trace and Zoom to approximate its root.

(c) Clear and Reset

Note: This problem is used as an example by Borland Software (makers of Turbo Pascal etc.) in their advertisement for Eureka, a new software package used to solve equations. (Ans: 2.07)

PROBLEM #7

Show that the composite of a function and its inverse is the identity function.

(a) Graph: $y = \ln(e^{x})$. Use real type exponent here.

(b) Graph: $y = e^{\ln(x)}$. Use real type exponent here.

(c) Question for discussion. Why the limited domain on b?

(d) Clear and Reset

PROBLEM #8

(a) Graph: $r = 6 + 6\text{Cos}(\theta)$

(b) Graph: $r = 6 - 9\ \text{Cos}(\theta)$

(c) Trace

(d) Clear and Reset

PROBLEM #9

(a) Change limits on θ to $\theta \in [0,7\pi]$

(b) Graph: $r^2 = 81\text{Sin}^2(5\theta/7)$

(c) Clear and Reset

PROBLEM 10

(a) Graph the parametric equations

$$x = 4\text{Sin}(t) + 4$$
$$y = 8\text{Cos}(t) - 2$$

(b) Trace

(c) Graph: $x = 4$ ($x = F(y)$)
$y = -2$ ($y = F(x)$)

(d) Clear and Reset

The following are other problems you might like to try on your own.

PROBLEM #1

(a) Graph: $y = 1/2\ x^2$

(b) Graph: $y = -x + 4$

(c) Graph $y = 1/2\ x^2 + x - 4$

(d) Graph: $x = 2$

(e) Graph: $x = -4$

What does this have to say about solving

$1/2x^2 + x - 4 = 0$ and $1/2x^2 = -x + 4$?

(f) Clear and Reset

PROBLEM #2

(a) Graph the following polar graphs:

$r = 10\text{Sin}(4\theta)$
$r = 8\text{Sin}(4\theta)$
$r = 6\text{Sin}(4\theta)$
$r = 4\text{Sin}(4\theta)$

(b) Regraph all equations

(c) Clear and Reset

PROBLEM #3

(a) Find $\lim_{x \to 0} \text{Sin}(x)\ /x$

(b) Use Trace and Zoom

(c) Clear and Reset

PROBLEM #4

(a) Find all solutions for x of the following

$1/4(x - 3/2)^2 - 6 = [x/4]$

Note: [x/4] is the greatest integer in x/4.

(b) Use Trace

(c) Clear and Reset

PROBLEM #5

(a) Graph: $y = (2x - 6)(x + 3) / (x^2 - 25)$

(b) By graphing appropriate graphs draw all vertical and horizontal asymptotes of the graph above.

(c) Clear and Reset

PROBLEM #6

(a) Change axes limits to the following:

$x \in [-5,5]$
$y \in [-10,4]$
X Tic = 1
Y Tic = 2

(b) Graph: $f(x) = [1 - x^2]$

(c) Does Limit f (x) exist?
$x \rightarrow 0$

(d) Does Limit f(x) exist?
$x \rightarrow 1$

(e) Clear and Reset

PROBLEM #7

(a) Graph: $x^2 + y^2 = 36$ and $(x - 9)^2 + y^2 = 9$

Hint: Use Conic - Circle from Graphing Menu.

(b) Draw all common tangent lines to these two circles.

Hint #1: There are three of them and the common internal tangent is easy to draw.

Hint #2: The two common internal tangents are more difficult to determine. Consider the diagram below and use the similar triangles to find the coordinates of intersection between the larger circle and the common tangent.

I hope that you have enjoyed your encounter with the Omnifarious Plotter and hope that you will use a graphics program in your own classes in the future. A graphing program can aid students in visualization, understanding, and discovery. It can also aid the teacher in the presentation and reinforcement of many of the high schools mathematical topics.

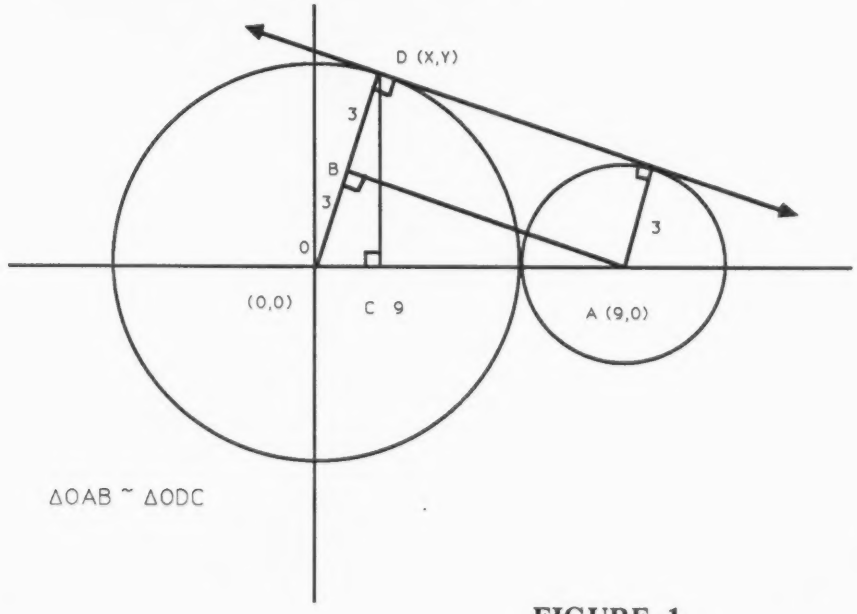


FIGURE 1

MONITORING STUDENT PERFORMANCE AND MANAGEMENT OF INSTRUCTION: MATHEMATICS, READING, AND LANGUAGE ARTS

By: Albert Erlebacher & John W. Wick, Northwestern University, Evanston, IL

We wish to describe here three diagnostic achievement tests available from Lagniappe Testing, Inc. There are five assumptions behind these testing programs:

1. Frequent monitoring of students' performance through testing is necessary for efficient instruction. Instructional time should not be used to teach what students already know. Conversely, students should not be advanced to levels for which they are not prepared.

In reading instruction, inefficient teaching results when the material is unchallenging and too simple for a student to learn new techniques. Nor can we expect learning to occur when the material presented to students is beyond their functional reading level and comprehension ability. In substantive areas such as mathematics and language arts, students do not have an opportunity to learn when they are "taught" material that has already been mastered. That can occur when the teacher continues teaching past the time when students have already mastered a topic or when, grade after grade, teachers "introduce" a concept, oblivious of what has gone on in previous classes. On the other hand, inefficient use of instructional time occurs when a teacher assumes that students are prepared for new material based solely on reports from their previous teacher that the prerequisite material has been taught.

2. Frequent testing to monitor students' performance can help achieve efficient instruction. At the beginning of the school year students ought to be tested to determine whether they have mastered that material considered prerequisite to material to be introduced. They also ought to be tested to determine whether they already have mastered the material that is to be taught. Then as the class progresses, frequent testing should determine when concepts have been mastered so that the teacher may go on to new material. For reading, testing can determine the functioning grade level for each student so that material of the proper difficulty can be given to each. Occasional testing will then determine when a student is ready to graduate to more difficult material.

3. Efficient instruction requires school-wide planning. Each subject area can be divided into a large set of objectives. Within a school a committee of teachers and/or the principal should identify for each objective, the grade of initial instruction and the grade when mastery is expected of most students. This identification will then determine the instructional flow in the school. Frequent monitoring of student performance, along with grade-by-grade reports, can help fine-tune the instructional program.

4. Teachers are most willing to test their students in order to achieve efficient instruction. However, they do now wish to spend much time creating appropriate tests, grading, and record keeping.

5. Computers are the perfect vehicle for implementing the kind of program suggested here. They allow flexible testing appropriate for a particular school's instructional program (and not dictated by a publisher's format). Computers also achieve near-instant scoring and allow for efficient record keeping. Finally, the data collected can be easily summarized by the computer for classroom-specific or school-wide reporting.

THE TESTS

The tests described here are intended to be part of the instructional program--not the evaluation program. They are not designed to be competitive with standardized tests that compare a school's outcome to national norms.

Diagnostic Achievement Test: Mathematics. Prior to any student being tested, the teacher customizes the test for the specific school or even the specific classroom, by choosing up to 33 objectives to be tested from a list of 178 objectives. This selection is transferred to the testing program disk so that all students in a class are tested with the same items. Students, seated at a computer, are tested individually at their own speed. Each objective is tested with three items and mastery is assumed if the student gets all three items correct. When the entire class has been tested, a class list is furnished, within minutes, that ranks the students in the order of their performance and ranks the objectives in order of their difficulty for this class. The list then shows for each student the number of correct answers for each objective. A glance at this list easily shows which of the objectives have been mastered by most of the students and which require more work. One also gets the notion whether the entire class ought to be instructed on generally nonmastered objectives or whether selected students ought to be brought together for specific instruction. A second print out shows a frequency distribution of the class, indicating, for each objective, the number (and percent) of students who have mastered it. Finally, a print out is furnished which lists each tested objective, one-by-one, along with the names of those students who did not demonstrate mastery of the objective.

Diagnostic Achievement Test: Language Arts. The main part of the language arts test tests students on capitalization, punctuation, and grammar. These are tested in the context of short passages, approximating what is expected when students proof read their own work. A second part of the test, tests students on terms and definitions. Selection of the objectives from a list of 176, the testing of students, and class summaries are similar to the mathematics test, though the three items testing a particular objective are not generally presented contiguously.

Diagnostic Achievement Test: Reading. This test differs from the other two in that it is not objective-driven. Instead, it is performance-driven. When the student's grade is entered, the computer selects and presents a passage of appropriate difficulty. The student then answers from five to eight questions concerning the passage. Based on the student's answers, the computer calculates a functioning reading level and selects the next passage to match that level in difficulty. This process goes on until the student has been tested with six passages. This adaptive testing process results in an accurate assessment of the student's functioning reading level and achieves this goal without the student having to read material that is clearly too simple and unchallenging or too difficult and beyond comprehension. The technique could, of course, not be used without the aid of a computer.

Minutes after the entire class has been tested, the computer presents a class list with the students ranked according to their ability and indicating, for each student, the functioning reading level along with some other diagnostics. If the teacher has so opted, the computer makes suggestions for reading group composition. An error analysis indicates the kind of questions on which the student has made errors and the type of errors made.

INFORMATION MANAGEMENT

With the management of instruction in mind, students will very likely be tested several times per year. This results in a large amount of information that also must be managed. The software is up to that task. Any time during the year, the teacher may get a year-to-date summary of a class's performance. Since the principal will want to know how the instructional program is working, a building level summary of year-to-date performance is also available. This entire process achieves a monitoring program that begins with the student and ends with the principal.

An added bonus of the software for the mathematics and language arts tests is that it identifies a large set of objectives. If the school has not yet listed objectives for itself, this list certainly can serve, at least, as a starting point. In addition, though each school should determine for itself the "grade of initial instruction" and "mastery grade" for each objective, Lagniappe Testing's software furnishes "default values" based on a sample of schools who have made such a determination. These values can serve as a starting point for curricular planning in those schools that have not yet created such a plan.

SOME DETAILS

1. The software is written for Apple IIe computers with 80-column card installed, monochrome monitor, and one disk drive.

2. All the programs are menu-driven, easily understood, user-friendly, and well documented. Checks are built in at various points to make operator errors near impossible.

3. Inquiries concerning these tests can be made to: Lagniappe Testing, Inc., 953 Oak Drive, Glencoe, IL 60022. Phone: (312) 835-2440

"BYTES" OF APPLE'S B.A.S.I.C. LANGUAGE GRAPHICS AND ADVENTURE STORIES
Grades 3-6

By: Beverly Fink, Des Plaines CCSD #62, Des Plaines, IL

The philosophy of my computer programming units supports a hands-on, learning-through-doing approach.

Students know in advance that they will be responsible for creating and debugging their own programs and will not be using any commercially sold software.

This fosters a sense of pride in accomplishment, better care of the equipment, and an appreciation of the more complex programming techniques used by programmers of commercial software.

YUM (grade 3) - an introductory programming unit.

Student Learner Objectives:

- to demonstrate proper care and handling of CPU, monitor, disk drive, and own disk.
- to use keyboard properly especially the RETURN key, CTRL key, arrows, and space bar. ("Touch typing"/keyboarding is the responsibility of the homeroom teachers through commercial software.)
- to demonstrate correct use of these commands: NEW, HOME, RUN, LIST, SAVE, LOAD, CATALOG, DELETE.
- to write original programs using these terms/symbols: line #, PRINT, GOTO, INVERSE, FLASH, NORMAL, ?, quotation marks.
- to analyze and debug their programs when SYNTAX ERROR is displayed.
- to design original flow charts with at least three decision boxes.
- to write original programs demonstrating the function of commas and semicolons outside of quotation marks.
- to write original LET programs,
- to demonstrate the use of the computer as a calculator with IF-THEN statements.
- to design original PRINT pictures.
- to demonstrate the proper use of the printer when printing out original programs.

YUMMY (grade 4) - emphasizes low res graphics.

Student Learner Objectives:

- to demonstrate initializing own disk.
- to seek out different programming languages and their functions.
- to demonstrate conversion of decimal numbers to binary.
- to synthesize previous programming terms and skills when writing more complex programs.
- to analyze sample program listings and predict how they will run.
- to write original FOR-NEXT programs.
- to distinguish between text and graphic screens.
- to design and print out detailed original low res graphics incorporating these terms: GR, COLOR, PLOT, HLIN, VLIN, REM, FOR-NEXT.
- to debug all original programs.
- to apply line editing techniques properly.

YUMMIER (grades 5/6) - emphasizes synthesizing keyboarding, programming, and writing skills into original adventure programs.

Student Learner Objectives:

- to design a branching flow chart for a published "Choose Your Own Adventure" type story.
- to demonstrate a branching flow chart for an original adventure story.
- to demonstrate the correct use of these terms/commands: INPUT, (STRING) VARIABLES, GOSUB, HTAB, VTAB, ON-GOTO.
- to synthesize new computer terms and commands and good writing skills into an original adventure story.
- to debug and print out original program.

YUMMIEST (grades 5/6) - emphasizes hi res graphics and basic animation techniques.

Student Learner Objectives:

- to explain differences between low res and hi res graphics.
- to design an original hi res graphic.
- to write a program to execute original graphic.
- to analyze sample low res animation programs.
- to write an original low res animation program.
- to debug all original programs.
- to design an original detailed hi res graphic using the DRAW program.

I have been instructing gifted students in these classes for four years. It is unfortunate that we have only one quarter to devote to programming. The students thoroughly enjoy the challenges and deal with the frustrations of being successful programmers. These classes inspire many of them to take summer and weekend computer classes and many have aspirations of becoming professional programmers.

MICROCOMPUTER-BASED INFORMATION SYSTEMS FOR EDUCATIONAL MANAGEMENT

By: Frederick Frank, Muriel Mackett, and Peter Abrams, Northern Illinois University, DeKalb, IL

> Computers and the types of information systems that computers make possible represent new challenges to those who wish to find ways to ensure that the machines we build are well suited to human use. ... They hold the promise ... of extending our intellectual resources much as the Industrial Revolution extended our physical capabilities. (Nickerson, 1986, p. 7)

The application of computers to educational management, while holding tremendous potential, is still a fairly new endeavor. In particular, the uses of microcomputer technology in educational information systems may well be in infant stages compared to the revolutionary potential noted by Nickerson. The workshop session presented by the authors at the conference on "The Role of the Computer in Education VII" focused on the development of integrated microcomputer-based information systems for educational management.(1) First, background information on the development of the microcomputer technology which supports integrated information systems was provided. The Achievement Management Program (AMP)™, part of the larger Integrated Information System for Program Development (INSYS)™ developed by the authors(2), was described as an example of such an information system. Then, an extended hands-on session was conducted to introduce workshop participants to the microcomputer technology and the applications which support integrated information systems, as exemplified by the AMP. Case examples of district applications of the AMP and discussion of issues related to implementing microcomputer-based informations systems were integrated into the hands-on session.

MICROCOMPUTER AND INFORMATION SYSTEMS

"Integrated microcomputer-based information systems for educational management" refers to the broad notion that microcomputers can be used to track data on many different aspects of school functioning and to support integration and analysis of that data (statistically and qualitatively) to generate a powerful information base for educational decision making and accountability. At minimum, well-developed integrated information systems should include data pertaining to student management, personnel management, and information management. (3) This capability represents a fundamental advance over what has been possible in the past and is a significant departure from more standard administrative microcomputer applications typically in use today. Key developments in mainframe computer and microcomputer technology and software have made this advance possible.

As recently as a decade ago, the option for school administrators to use microcomputer technology as a school management tool was virtually non-existent. Almost all of the administrative tasks, for which there are now microcomputer applications, had to be carried out "by hand." Even with the emergence of mainframe computer applications in the school setting beginning in the 1950's, traditional practices continued in the great majority of instances. Mainframe systems offered comparatively spectacular advances over "by-hand" systems. But their relatively high cost, complexity, and difficulty in accessing has resulted in their being used primarily as technologically-based versions of the old "by-hand" systems. Further, mainframe systems have continued to be thought of and used in

connection with highly defined and routinized data processing activities carried out at the central office level, such as payroll, attendance, basic student demographic data and standardized achievement test scores. Certainly, mainframe systems have become significant management tools at the central office level, but their functions are often removed from educational program decisions and very often are not of direct help at the building level.

The development of the microcomputer has created a whole new set of options for carrying out a number of the tasks of school administration. Recently, there has been a great deal of general information available to public school administrators about potential uses of microcomputers in school management. Trade journals, the popular press, mail solicitations, journal articles, conferences and workshops -- and no shortage of consultants -- have offered numerous opportunities for administrators to use several different software programs designed to aid in or accomplish administrative tasks. Microcomputer capability has revolutionized the potential for technology to be used as an essential school management tool -- at school building and central office levels -- and the possibilities are growing at a faster rate than new applications can be applied in the school setting.

Most of the available software programs, however, have been single-purpose programs, designed for carrying out specific, discrete administrative tasks such as scheduling, record keeping, budgeting, or school accounting. Although some software programs offer a set of such options as part of a single package, they have remained essentially single-purpose programs based on separate file structures. In other words, scheduling, record keeping, budgeting, school accounting, and other such administrative tasks are in most instances carried out as separate functions, even when they are part of the same microcomputer software program.

It is clear that microcomputer applications for school administration are proliferating and offering many benefits for school operation. It is equally clear, however, that these applications have not reached their full potential to produce integrated information systems for educational management which will allow tracing, integration, and analysis of student, personnel, instructional, fiscal, organization, and information management data. Although not widely recognized, at least among school personnel, the technology to do so does exist. The AMP has been developed and made available to school districts in order to apply the available technology to meet the need for integrated information systems. The AMP appears to be somewhat unique in providing this essential administrative application in the educational arena.

THE ACHIEVEMENT MANAGEMENT PROGRAM

The AMP is a comprehensive data management program developed to assist building principals and central office personnel to manage student achievement. The program is driven by: (1) comprehensive, frequent, and periodic data collection and analysis of factors related to student achievement; and (2) development and implementation of data-based pedagogical and achievement intervention strategies designed to address student achievement needs. The AMP uses IBM-compatible personal computers (PC's) and integrated spreadsheet and statistical software. The AMP is, at this time, designed to use Lotus (R) Symphony (R). (4)

The AMP has been created as a response to emergent educational needs and remarkable advances in the technologies related to information processing and analysis. Specifically, these educational needs center on addressing student achievement problems and increasing the effectiveness of building principals and teaching staff, as well as other administrators in terms of their ability to monitor, understand, and impact student achievement. To be

effective, educational personnel need good, timely information and plenty of it. Further, principals and other personnel need that information in a form which supports problem solving and subsequent action. Advances in technology, particularly the advent of the PC configured with expanded memory, have dramatically changed the capability of the principal to obtain, maintain and process information. The recent appearance of sophisticated, integrated spreadsheet software programs such as Lotus (R) Symphony (R) have revolutionized data display and analysis. Simply stated, there is a great need for all school administrators -- but particularly building principals -- to have much more good information about how students are achieving and now, for the first time, there is a cost effective way to get this information and work with it.

THE HANDS-ON WORKSHOP

Using the AMP as a framework, the hands-on session was designed to introduce participants to the concept of integrated information systems, to the hardware and software technology which makes such systems, not only feasible, but essential and desirable educational management tools, and to the spreadsheet and statistical manipulations which the user must master in order to apply the system. Conducted over approximately a six-hour period, the session covered the following main items. Illustrative materials (hard copy) corresponding to each item were integrated into the session.

* Single year and multi-year spreadsheet information systems.
* Single year spreadsheet for a single school class.
* Single year spreadsheet for a class with statistics.
* Multi-year spreadsheet for a school, alphabetical.
* Multi-year spreadsheet for a class.
* Interpretation of single year statistical (SYSTAT)™ (5) variable names.
* Interpretation of multi-year statistical (SYSTAT)™ variable names.
* Statistical application options.
* Describing categorical data.
* Describing continuous data.
* Prediction.
* Microcomputer applications for school improvement.

Integrated into the session were case examples of AMP applications in a school district and discussions of major issues related to administrative applications of microcomputers. Issues discussed included data accuracy, security, access, and confidentiality. Further information about the overall content of the session, about the specifics of the hands-on workshop, and about the Achievement Management Program and INSYS is available from the authors.

REFERENCE NOTES

(1) Mr. Kenneth Cull, Principal of Frost Junior High School in Schaumburg, IL School District 54, was also a presenter at the workshop. Mr. Cull directs the AMP in District 54 and has worked with the authors as a consultant for SASA.

(2) The Achievement Management Program, AMP, The Integrated Information System for Program Development, and INSYS are registered trademarks of School Administrator Service Associates (SASA), 412 Joanne Lane, DeKalb, IL 60115.

(3) Elaborated treatments of the authors' conception of integrated information systems for educational management appear in a number of articles. Examples include: F. Frank, M. Mackett, P. Abrams, and J. Nowakowski, "The educational utility and educational administration and management" and M. Mackett, F. Frank, J. Nowakowski, and P. Abrams, "Preparing educational administrators" in D. Gooler, The education utility: The power to revitalize American education and society, Englewood Cliffs, NJ: Educational Technology Publications, 1986; M. Mackett, F. Frank, P. Abrams, and J. Nowakowski, "Computers and educational excellence: Policy implications for educational administration," in D. Griffiths, Leaders for America's schools, San Francisco: McCutcheon (in press).

(4) Lotus (R) and Symphony (R) are registered trademarks of Lotus Development Corporation, 55 Cambridge Parkway, Cambridge, Mass., 02142.

(5) SYSTAT ™ is a trademark of SYSTAT, Inc., Evanston, IL.

REFERENCES

Nickerson, R. S. (1986). Using computers: The human factors of information systems. Cambridge, Mass., MIT Press.

BEYOND SOFTWARE EVALUATION

By: Mrs. Bobby Goodson, Education Technology Consultant, Sunnyvale, CA

Before we look ahead to where we think we should be going in the evaluation and selection of educational software, it might be worthwhile to look back and remember where we have been. One way to do that is to look at some of the guidelines we have had in the past.

The first major effort to establish standards for software evaluation came from the National Council of Teachers of Mathematics in 1981. Those guidelines, in retrospect, were ahead of their time, especially considering that we were using computers with 48 - 64K memories, cumbersome storage capabilities, limited graphics, etc. They directed us to think about instructional groupings, the uses of the program, the teacher's and the student's point of view, the instructional and motivational style of the program, and the social characteristics.

In 1982, MicroSIFT, at the Northwest Regional Educational Laboratory, produced an evaluator's guide with another set of guidelines. This evaluation process emphasized the content - the accuracy and educational value, the instructional value - the purpose, appropriateness, and effectiveness, and the technical quality - the level of support and ease of use. That same year, Ann Lathrop published her Terrible Ten - ten reasons for rejecting a piece of software. How long has it been since you have seen software that - gives an audible response to every answer, rewards failure more than success, contains technical problems and inaccurate information or scrolls the screen display? We have come a long way in five years!

Ann Lathrop and I wrote Courseware in the Classroom in 1983 and we presented a case for the well-planned use of computers within the regular classroom. Our evaluation guidelines stressed the truly creative uses of the computer, applied to recognized instructional objectives. We expressed concern for the type of response expected of the student and the computer response in return. The quality of the screen formatting, the type of motivational devices and the general technical quality of the software we felt also needed close scrutiny.

Finally, in 1985, the state of California established guidelines for educational software in California schools. [1] These guidelines outline both the minimum standards and the more desirable qualities we expect to see in educational software. They describe the qualities of three categories of software - the essential attributes (minimum standards), the desirable attributes, and the indicators of excellence. Each of these categories addresses the educational content, the technical features and the support material presented with the software used in our schools.

When we have come this far in determining our standards for educational software, what more can we do? With so much educational software on the market now, we need to consider our review and selection process. As more and more software is being used in individual classrooms, we need to realize that in the final analysis software evaluation and selection is a personal thing, and classroom teachers need help in making wise decisions.

The review process can be carried on at several levels, relying on input from a variety of sources. Popular journals furnish general descriptions of some software, while various commercial and professional organizations provide more detailed information.

Publications, such as TESS, the Educational Software Selector, published by EPIE [2] includes detailed information about more software than will be found in any other place. Resource centers around the country often do organized screening of software and provide information, such as the Educational Software Preview Guide [3] to assist teachers in the initial selection of software. These "third-party" reviews are very helpful in screening software for technical problems; such things as complicated operating procedures, inconsistent routines, factual errors, and the lack of support material.

However, the final selection of software for classroom use needs to be made through a personal evaluation process by the individual teacher. In spite of all the objections about the time it takes, teachers make most other choices of supplementary material used in their classroom, and who else can judge whether the presentation fits their teaching style, whether the material is suitable for their specific class, or whether that particular application is appropriate for the lesson they are teaching? In other words, the final choice needs to be made with consideration of the students and the curriculum, not just quality alone.

If technology is really to become a vital part of the teaching process, it must take its place with all of the other aids the teacher has to meet the needs of each student. Just as there is a good time to use a film or a video tape to illustrate a point or manipulatives to explore a relationship, there are times to use a computer program to expand the lesson. Software needs to be selected to meet the needs of a particular subject area, enhance a special lesson or illustrate a difficult concept. It needs to fit into a lesson plan, with the same preparation as any other material and follow-up activities to tie it all together. When used in this way, we will begin to see how great the possibilities truly are.

In an effort to go beyond the evaluation of software for quality only, the California State Department of Education established a project, Technology in the the Curriculum (TIC), to identify material that was of both high quality and consistent with the curriculum as taught in California schools. During the first year, individual projects considered material in Language Arts, Mathematics, Science and Social Studies. During the second year, Foreign Language and the Visual and Performing Arts are being investigated.

The evaluation process for the TIC projects is three-pronged, considering the application of technology within the subject area and the appropriateness of the material related to the curriculum, as well as the quality of the material. To meet these needs, the staff of each project started with a philosophical statement, or position paper, to establish the appropriate role of technology, both computer and video, in that particular subject area. It had already been decided that only software meeting the desirable or exceptional software guidelines would be considered. Finally, the establishment of the final criteria involved the refinement of the K-12 curriculum framework in each area into a form that software evaluators could use as they established the suitability of each program.

With all of these considerations, the evaluation process took on a very different feel. It meant that very good software was rejected because it did not represent an appropriate application or the specific curriculum objectives. Good software that really supported the curriculum had to be dropped because the reading level or presentation style did not match the level at which those objectives are presented. It meant that more software sources were explored as the search for appropriate material proceeded. The result was a group of approximately five hundred pieces of computer software and several hundred video titles that met a very special set of criteria.

This material, the position papers, the curriculum scope and sequence, and the evaluative descriptions of the identified programs were compiled into a series of resource guides.[4] We knew, however, from experience that supplying another list of software to the schools was not, in itself, going to make a significant difference in technology use. Teachers still needed help in using the new material. In an attempt to meet this need, a series of model lesson plans were developed for each subject, in each major grade grouping. These lesson, or more often unit, plans focused on particular parts of the curriculum that illustrated especially appropriate uses of technology. Wherever possible, these lessons incorporated both computer and video material. They demonstrated the process of planning a normal, necessary lesson taught in every school, but presented in a new way, or possibly enriched to encourage new depths of understanding.

The first set of four guides and a new data base, the DataRelator, developed for the project, were delivered to every school in California last Spring. The next set, Foreign Language and the Visual and Performing Arts will be ready this Spring. The process of updating the original guides with the new material that has come out in the past year is underway. We must now answer some serious questions.

* Has this new material made a significant difference in the use of technology in California Schools?

  It is being well used by some teachers, those who were ready to try new ideas and had access to the necessary material. It has greatly improved the planning process and the software purchasing habits in many schools.

* Is this enhanced evaluation process a valid technique for improving the quality of material being used in the classrooms?

  The identification of the role of technology is an important part of the planning process for a school. It enables the staff to focus their efforts and develop their skills. It must be remembered, however, that the uses of technology are evolving and this portion of a plan needs to be reconsidered periodically!

  The addition of curriculum considerations to the evaluation process is well worth the added effort. It seems to be very helpful in assisting teachers to focus on the specific uses of technology and directing purchases to the most relevant material available.

* What needs of teachers must still be addressed, if they are to make more and better use of technology in their classrooms?

  Even though the TIC material was delivered to the schools with a half day of training to introduce the material, it is obvious now that more training is needed. The incorporation of technology into the daily lesson is a new and undeveloped skill for most teachers.

The TIC project was a major and extremely worthwhile undertaking. While each state has its own unique curriculum, these guides based on the California curriculum present a model that others should certainly adapt. The concept of software review related to curriculum takes additional training for the reviewers, but the end results are well worth the effort. The integration of technology into the day to day learning process requires additional training and support for teachers and this seems to be one positive way of providing that support.

(1) GUIDELINES FOR EDUCATIONAL SOFTWARE IN CALIFORNIA SCHOOLS,
Educational Technology Unit, California State Department of Education, 721 Capitol Mall, Sacramento, California, 95814-4785

(2) TESS, THE EDUCATIONAL SOFTWARE SELECTOR, ($59.95)
EPIE Institute, P.O. Box 839-T, Water Mill, NY 11976

(3) THE EDUCATIONAL SOFTWARE PREVIEW GUIDE, ($8.00)
International Council for Computers in Education,
University of Oregon, 1787 Agate Street, Eugene, OR 97403-9905

(4) TECHNOLOGY IN THE CURRICULUM RESOURCE GUIDES in Language Arts, Math, Science and Social Studies are available ($95 for a complete set) from Publication Sales Office, California State Department of Education, P.O. Box 271, Sacramento, California, 95802-0271
Complete sets, including Foreign Language and the Visual and Performing Arts will be available for $140.00 later this year.

A BULLETIN BOARD SYSTEM FOR SCIENCE TEACHERS

By: Brad Goral, New Trier High School, Winnetka, IL

The teaching of high school science requires teachers to stay current in their fields. Traditionally, this has been done through formal classwork, in-services, and conferences, all of which can be effective, as most teachers will attest. There are times, however, when these methods are unsuitable. Have you ever seen a bulletin on some conference or another and wanted to go, but didn't because of the inconvenience? Or do you ever want to talk to colleagues about an interest that you have at that particular time, and find that none of your department members really shared the same interest? Or are you the kind of person who just likes to communicate about things more than most?

The commercial bulletin board programs for computers open another avenue for effective communication among people of similar interests. These systems also provide a facility for quickly transferring large amounts of text, or computer programs. The convenience and speed of this medium makes it a very powerful means of remaining "up" on things.

New Trier High School is presently operating a bulletin board system for science teachers. Such a system, regardless of its theme, becomes useful to callers only as a variety of people contribute to the dialog and file banks that make it up. While such a system will probably never become really busy, it can certainly provide a means of communication for groups of teachers, maybe small or maybe relatively large, that meets a real need. Secondarily, or perhaps most importantly for some, the system can provide a large source of teaching materials of all sorts which should prove useful to callers.

The Junction (as this BBS is called) solicits file contributions from science teachers in the Chicago area, and it si certainly hoped that some will find the system an excellent means of engaging in dialog with other teachers.

Here is the menu of files which are presently available. All material is, of course, in the public domain. Each file, which is documented briefly on this menu, is discussed more extensively in some other files that are dedicated to that purpose.

Download area: SCIENCE
Workspace: /JUNCTION/DLOADS/SCIENCE/

FILENAME	TYPE	COUNT	TIME	DESCRIPTION
029 Planet Distance	TXT	00002	02Min.	Apple Prog Calc Rel. Distances
028 Computer.Music	TXT	00000	05Min.	Apple Prog Demo Freq & Pitch
027 DNA	TXT	00000	19Min.	Apple Prog General Intro to DNA
026 Tongue.Shapes	TXT	00000	01Min.	Apple File (Binary)
025 Tasting.Lab	TXT	00000	10Min.	Apple Prog Areas of Tongue
024 Calories	TXT	00000	08Min.	Apple Prog Calcs Cals Burned by Person
023 Cal.Bal	TXT	00000	03Min.	Apple Prog Compares Cal In/Output
022 Cum.Grade.Analy	TXT	00001	06Min.	Apple Prog Graphs Student Progress
021 Gradebook	TXT	00000	29Min.	Apple Prog for Keeping Grades

CONTINUED-------

FILENAME	TYPE	COUNT	TIME	DESCRIPTION
020 Libraries	TXT	00000	02Min.	Online Catalogs of Library Holdings
019 Eliza	TXT	00000	14Min.	Apple Prog Computerized Psychoanalysis
018 Calendar	TXT	00000	10Min.	Apple Prog Any Month Any Year
017 School.Board	TXT	00000	01Min.	BBS Devoted to Computing For Educators
016 Photo.Reaction	TXT	00000	02Min.	Demo of a Photo-Induced Reaction
015 Chem.Pursuit	TXT	00004	01Min.	Question Files for Apple Prog
014 Chemical.Search	TXT	00000	19Min.	Apple Prog On Phy Properties of Subst
013 PTD.Lab.9	TXT	00000	08Min.	Apple Prog Lab 9 Davis, MACNAB
012 PTD.Lab.25	TXT	00000	05Min.	Apple Prog Lab 25 Davis, MACNAB
011 PTD.Lab.18	TXT	00000	13Min.	Apple Prog Tutor Lab 18 Davis, MACNAB
010 Program.Lab.29	TXT	00000	04Min.	Apple Prog Lab 29 Davis, MACNAB
009 MGO.Lab	TXT	00000	06Min.	Apple Prog Calc Empir Formu (MG-->MGO)
008 Balloon.Trick	TXT	00002	01Min.	Fun Demo Using Balloons
007 Imploding.Can	TXT	00004	01Min.	Demo of the Effect of T on P of a Gas
006 Make.Solutions	TXT	00000	01Min.	Lab on Molarity of Solutions
005 Magic.Sand	TXT	00000	01Min.	Demo of Hydrophobic Material
004 Endothermic.Rex	TXT	00000	01Min.	Demo of Impressive Endo-thermic Rex
003 CMPD.Synthesis	TXT	00000	02Min.	Lab on Inorganic Syn Emphasiz Stoich
002 Exploding.Can	TXT	00001	02Min.	Demo of a Combustible Mixture
001 Catalog	TXT	00000	01Min.	Info on Source of Public Domain Softwa

TEACHING SOMETHING USEFUL IN 1000 MINUTES OR LESS

By: Michael Gordon, Maercker S.D. #60, Clarendon Hills, IL

INTRODUCTION

Situation

For all their glamour, computers are still the stepchildren of educational technology. Even the most enlightened school district may treat computer literacy as a supplement to the curriculum. As such computer instruction may be offered as an elective, or perhaps the time for computer instruction may be borrowed from some other subject. At the elementary level this may mean fewer minutes of Science or Social Studies of whatever; at the junior high level, computers may time share a period of a student's day with Art, Home Economics, Industrial Arts, Music, or all of the above. That there is precious little time to give instruction about computers is not necessarily bad; that there is precious little time to give instruction with computers is regrettable.

Philosophy

By far, most computers (personal computers anyway) are not used by people who are exploring new territories in computer science; most computers are used to solve a problem by applying the computer as a tool to improve or simplify a task which otherwise must still be done. Writing with a word processor, keeping records with a database, and manipulating numerical information with a spreadsheet are most common. If being literate is being well versed in the subject under consideration, then being computer literate means knowing how to apply the computer as a tool to assist in the solving of problems.

Teaching computer literacy means teaching how to use the computer in the ways that computer literate people use them. That means then that we should be teaching the applications of the computer such as word processors, database programs, and spreadsheets to improve the study of the other subjects we are now teaching. That is not to say that teaching programming in any form has no place. It means that our teaching of the use of the tool must reflect the ways the tool will actually be used. Concerning cars, we teach students to drive them safely, and if we do the job completely, we teach enough of how a car works to understand what constitutes appropriate maintenance; we do not teach all students to be auto mechanics. Similarly, we should teach students enough about the skills of programming so that they may appropriately use and maintain their computer tool. Like auto mechanics some may wish to pursue programming, but all need not learn that level of the craft.

Challenge

The challenge is how to do this given the amount of time which has been stolen from the other subjects, and do it in such a way that the teaching of the other subjects is enhanced and not impaired. In my school district, instruction begins in the first grade and continues each year through the eighth grade using time stolen from some other subject. For computer instruction, amounts of time ranging from three periods of 25 minutes (75 min.) to five weeks at 40 minutes per day (1000 min.) are used depending on the grade level. How can anything meaningful be taught in such a fragmented fashion?

The Gold Star Approach

When I was little, my teachers used to give gold stars each time I successfully completed a task. It felt wonderful to complete the assignment and receive the reward. After a time I didn't even need the stars; it felt good to complete an assignment; that feeling never went

away. We can help our students by assigning them manageable tasks which they can attempt, complete in a reasonable time, and receive a visible product as a reward (a few gold stars never hurt either). Word processing, database, and spreadsheet applications lend themselves to this approach. Using the appropriate tools, even pre-programming computer literacy can be effectively taught in a limited amount of time.

IMPLEMENTATION

In the discussion that follows, I would like to give you a taste of how I implement this philosophy. At each grade level I try to introduce both skills and concepts which will teach and reinforce in the students the idea that the computer is a tool which they can manipulate in order to better accomplish tasks assigned to them. While I will not describe each level in detail, I have listed the time allotted to each skill/level.

Word Processing Skills

Word processing is in some ways the most useful of the computer tools for teaching. It reinforces the idea that the student controls the computer; it allows him to manipulate the product of his work; it has application as a tool in most subject areas.

Third Grade: 450 minutes/2

Using Logowriter, we introduce the third graders to two concepts: (1) that their is a relationship between the commands that they give the computer and the actions taken by the computer (explained in more detail below), and (2) that they can enter and process information (text and pictures) to their own specifications; each child works on creating a card (Birthday, Mother's Day, etc.).

Fifth Grade: 450 minutes/2

Search and Replace Math worksheets. An interesting application of word processing skills is the use of the Search and Replace function applied to the creation of math worksheets. Each child loads in a file containing problems such as:

1. !Name! has twenty-seven !fruits!. !Name! divides these !fruits! equally among three friends. How many !fruits! does each friend receive?

The file contains a number of such problems and a set of directions including:

1. Using the Find and Replace function:

    a) find !Name! and replace with a name of your choice.

    b) find !fruits! and replace with the name of your favorite fruit (plural).

2 - 8 ..... (Similar problems)

9. Print out a copy of the problems and do them for homework.

Few teachers look forward with anticipation to working on word problems with students, but with this approach the children often attack the problems with relish since "they made them up," and they make use of a valuable word processing function in the process. We know that they were doing the problems that we wanted them to, anyway.

Kids love mysteries. The very first day with the sixth graders, I tell them that they will be writing a mystery story as their project. I suggest that it will be a very dull mystery if they blurt out the whole story in one paragraph, and so perhaps they should see how other mystery writers do it. I ask them to read a mystery, long or short. Hardy Boys, Nancy Drew, and Encyclopedia Brown would be wonderful examples of good mystery books. There is always a run on these books in the library. In order to get ready, we need to learn how to type text into a word processor, I explain.

For a word processor, I chose FrEdWriter because of two features: (1) it has the ability to enter directions and other text in such a way that they will not be easily modifiable by the children and will not print unless desired; (2) it is free, and therefore we may make copies for use and distribute them to all of the children. Those who have Apple computers of their own obviously will have use for it, but we also make the school's computers available for preparing homework assignments for their classes.

This project is ready made for the gold star approach. There are some wonderful articles in Teaching and Computers which make teaching word processing this way easy. Two in particular, by Thomas Boudrot, facilitate teaching word processing in manageable bites (reverse pun intended) which he terms tasks. The first is found in the February 1986 issue entitled "The Secrets to Teaching Word Processing or the Legacy of Morris Q. Moneybucks;"; the second is entitled "Better Letters."

"Morris" is a series of activities centered around the legacy left by a rich man to his three somewhat greedy children. In task one, Morris leaves clues to hidden treasure planted around his mansion. The students have to load files with names like CLOCK, PORCH, and SINK, each time reading the clue which leads to the name of another file to load. Eventually, on the way to finding the hidden treasure in the last file, they have learned to load files and erase the file from memory.

In other tasks they learn to move the cursor, insert text, correct errors, and save files. Most importantly, while pretending that they are alternately each of Morris' three children: Horace, Doris, and Chloris, they write letters which express their own invention and have to correct their own errors on their way to a finished product.

Reinforcement is provided for all of these newly acquired word processing skills with a second series of tasks involving an apparently Junior High aged boy named Poindexter. Poindexter writes letters to his grandmother giving our students practice in correcting Poindexter's terrible spelling. The students also pretend to be Granny writing back to Poindexter giving him advice.

In addition, our students win an "opportunity" to be the personal "assistant" of a Hollywood star and discover that they will be doing his laundry and running his errands. In order to accept this choice assignment, they must write an acceptance letter and arrange with their teacher (by letter, of course) to keep up with their school work. They use all of the previously learned word processing skills and, in addition, get practice writing "friendly" and "business" letters.

Logo as a Programming Language

Even today, a number of years after its commercial introduction, Logo is viewed by many teachers as a simple to use drawing program. Even many of the enlightened abandon its use shortly after some "discovery" exercises on the drawing of geometric figures. The

truth of this claim is supported by the recent introduction of Logowriter by LCSI. Programmability has not been a strong promoter of the sale of Logo, and Logowriter is now promoted in their ads as putting "word processing, graphics, and programmability together in an accessible way," with programmability listed third. Logo is largely overlooked either as a programming language in its own right, or as a preprogramming educational language. Logo is, however, an ideal educational programming language for children because it not only has a simple threshold, but also allows structured programming style and possesses two characteristics of advanced languages: list processing and recursion.

Third Grade: 450 minutes/2

This is where I begin the development of the first programming skills. Using Logowriter and robot turtles, a connection is built between the commands typed and the turtle's actions. Using the lessons provided by LCSI, a fast paced preview is given of the capabilities of some of the basic Logo commands: forward, back, right, left, repeat, penup, and pendown. The children then move to the robot turtle and experiment with the commands.

A wonderful series of exercises may be built around "Turtle Shuffleboard." A shuffleboard game is drawn on a large piece of paper taped to the floor. Lines are drawn a whole number of centimeters from a starting line at assorted intervals in order to create point zones. Children first experiment with moving the turtle from the starting line in with the forward command in order to see how far the various numbers used as inputs will take the turtle. By carefully calibrating the turtle to move in centimeter units, the students begin to develop the ability to estimate distances in centimeters as well as becoming familiar with the forward and back commands. To make the game fun, teams play for points, but the opposing team gets to place the turtle anywhere within a starting zone so that distances must be estimated, not memorized. Similarly, a turtle version of roulette (or if you prefer, Wheel of Fortune) develops the ability to estimate angles.

Fourth Grade: 450 minutes

Experience in creating and using procedures is the next skill which must be developed. This is programming, however, the metaphor of a dictionary used in Logo instead of the traditional use of the term "programming" is a very important organizing idea. It is easy for children to accept that they can teach the turtle new words to add to its vocabulary; just as importantly, if the children visualize the turtle using definitions it has copied from its dictionary in order to do the things that it does, it builds a foundation which will make the concept of recursion (one of the most powerful features of Logo) easier to teach.

Four concepts make an effective package for writing a procedure: (1) writing a procedure, (2) using one procedure within another procedure, (3) the total trip concept (360 degrees or multiple total turns bring turtle back to same heading), and (4) drawing regular polygons (using n turns of 360/n degrees).

Seventh Grade: 1000 minutes

Tic Tac Toe is an often used project with which programming skills are taught. The complexities of other languages limit how sophisticated an implementation one could reasonably expect to do with a whole class. With Logo it is possible to plan and execute a two player version of the game "from the top down". With the better students detection of a win along with the traditional line to show a win or "c" to show a "cats" or drawn game is quite possible.

It is even possible to take the students who are really interested and develop strategies for a one player game where the computer plays the other position.

This project incorporates the programming concept of "Top Down Design", reinforces the skills in the use of procedures, applies the concepts of "inputs" and "lists", gives practical application (or teaches) the use of Cartesian Coordinates, and if the advanced concepts are included, can show practical applications of recursion. Besides, it's fun.

Data Base Skills

Fifth Grade: 450 minutes/2

Eighth Grade: 1000 minutes/2

Spreadsheet Usage

Eighth Grade: 1000 minutes/2

Much as I would like to discuss the above topics, space will not permit. I cannot overemphasize, however, the equality of importance of data base and spreadsheet skills with those of word processing and pre-programming literacy. The important aspect of the presentations in each of these subjects is that they are task oriented and goal directed. Each student proceeds through a series of manageable tasks which lead him toward the goal of basic proficiency in the use of each of these computer based tools.

DIRECTION

Computers are becoming increasingly important in our world; the use of computers is becoming increasingly commonplace. This fact will eventually redefine the concept of literacy as we teach it in our schools. In order to address the problem of the competition for educational time, we must seek ways to teach useful skills in manageable amounts. With an increasing level of computer literacy with our students, we may also move to true application of the computers as educational tools within the other disciplines in which we are required to instruct.

INTRODUCTION TO APPLEWORKS

By: Bob Inglesby, Arlington Hts. S.D. #25, Arlington Hts., IL

At this session at the Conference, as participants entered they were able to immediately begin investigating Appleworks through the use of an interactive demonstration program designed for Apple dealers. Because the program is menu driven, it allowed for the predictable situation of some participants having some experience with one part of Appleworks, but not all of the program. Usually, it seems, the first portion to be learned is the word processor. From the main menu of the demonstration program (see addendum 1) people would choose only those parts of the program they wished to learn about.

Fortunately there was a computer for almost every participant. This allowed each one to self-pace and even review a completed section. During this time, I was able to circulate through the group and answer questions individually.

About half way through the session I handed out a worksheet I had used with eighth grade students to introduce the power of using the data base portion of Appleworks (see addendum 2). This worksheet makes use of two data base files from Computer Applications by Arthur Luehrmann. The first is information about the planets of the solar system, the other has data on the fifty states and Washington D.C. (See sample on addendum 3). By using existing data files, the worksheet emphasizes the power of the data base's sorting and searching capabilities rather than spending time on the procedures for setting up a data base file.

The final activity was a brief look at a sample spreadsheet file that computes car loan payments. By changing numbers for the principal, interest, and term of the loan, the monthly payment is found. Since the spreadsheet is set to recalculate automatically, the "What if?" game can be played. Questions like, "What if we have decided to not spend more than $400 a month, what interest rate will enable us to borrow a given amount of money for a given term?" can be answered by trying different rates and watching the payment change.

I chose these activities assuming the session participants would have a wide variety of backgrounds. In addition to the demonstration program, I chose activities for the data base and spreadsheet and not the word processor, because word processing seems to be the first portion stressed by most introductory books on Appleworks, and is most likely the one portion that participants would have already experienced.

ADDENDUM 1 - APPLEWORKS DEMONSTRATION MENU

	1. Introduction
Word Processor:	2. Writing Made Easy
	3. No More Messy White Stuff
	4. Getting It on Paper
Data Base:	5. Throwing Out the Shoebox
	6. Find What You Need When You Need It
	7. Instant Mailing Labels

Spreadsheet: 8. World's Largest Ledger Paper
9. What if you Change a Number?

Integration: 10. Cut and Paste

Type a number (1-10) then press RETURN:

ADDENDUM 2 - DIRECTIONS FOR DATA BASE

Boot the startup side of Appleworks, then put in the PROGRAM disk and press return, correct the date if wrong and press return again. From the MAIN MENU choose "Add Files". Choose #1 The current disk: Drive 1. When asked for the data disk, turn the disk over (data is on the back). Using the down arrow, go to the file called "Planets" and press return.

(1) In what order is the list?

Press the TAB key once.

You should be in the column labeled "Diameter".

(OA-means hold down the OPEN APPLE key found just to the left of the space bar then press the key following the dash). Now do OA - A. Choose #4 and press return.

(2) Now, in what order is the list?

(3) Which planet has 4 planets larger than it and 4 smaller?

(4) Which planets tend to have more moons, the 4 largest planets or the 5 smallest?

Press the ESC key until you are back to the MAIN MENU. Follow the steps you used to load "Planets", but this time get the file called "States".

(5) In what order is the list of states?

To see the rest of the list hold down the down arrow key until the computer beeps. To return to the top do OA- 1.

Press TAB to move to the column labeled "Admitted". Do OA- A. Choose #3.

(6) What 5 states entered the union during the 1900's.

To move a column to the left use OA- TAB. Move to the column labeled "Area". Do OA- A. Choose #4.

(7) List the 5 largest states by area.

Do OA- A in the column labeled "Population". Choose #3.

(8) How many states have a population less than one million? (Do not count Washington D.C.)

Do OA- F. Type Augusta and press return.

(9) Augusta is the capital of which state?

To find all the capitals that contain the word "City" do OA- F. Then type City.

(10) How many capitals have the word "City" in their name?

If someone wanted to find a state where the average temperature during January was not too cold (> 40) and in July it did not get too hot (< 80) you would do this: Do OA- R. Choose #5 Jan. Choose #2 is greater than. Type 40, press return. Choose #1 and, Choose #6 Jul. Choose #3 is less than. Type 80. Press ESC.

(11) List the possible states.

ADDENDUM 3 - SAMPLE OF "PLANETS" AND "STATES" DATA BASES FROM COMPUTER LITERACY

PLANET NAME	DIAMETER	DISTANCE FROM SUN	MOONS
Mercury	3100	36.0	0
Venus	7519	67.2	0
Earth	7926	92.9	1
Mars	4218	141.5	2
Jupiter	88700	483.4	13
Saturn	75100	886.0	10
Uranus	29200	1782.0	5
Neptune	27700	2792.0	2
Pluto	3000	3675.0	1

NAME	POPULATION	AREA	CAPITAL	JAN	JUL	ADMITTED
Alabama	3,959,000	51,609	Montgomery	48	82	1819
Alaska	479,000	586,412	Juneau	24	56	1959
Arizona	2,963,000	113,909	Phoenix	52	92	1912
Arkansas	2,328,000	53,104	Little Rock	40	82	1836
California	25,174,000	158,693	Sacramento	48	63	1850
Colorado	3,139,000	104,247	Denver	30	73	1876
Connecticut	3,138,000	5,009	Hartford	25	72	1788
Delaware	606,000	2,057	Dover	32	76	1787
Dist. of Col.	623,000	67		36	79	1791
Florida	10,680,000	58,560	Tallahassee	55	81	1845
Georgia	5,732,000	58,876	Atlanta	42	78	1788
Hawaii	1,023,000	6,450	Honolulu	72	80	1959
Idaho	989,000	83,557	Boise	29	75	1890
Illinois	11,486,000	56,400	Springfield	25	75	1818

THE SHOW AND TELL VISUAL PROGRAMMING LANGUAGE

By: Takayuki Dan Kimura, Washington University, St. Louis, MO

Show and Tell™+ (STL) is an icon-driven visual programming language. Since both programs and data in STL are pictorial, a program can be created solely by using a pointing device such as a mouse. A keyboard is needed only for entering textual or numeric data. STL is designed for school children and novice computer users. 'Keyboardless programming' is the main goal of the language. The name 'Show and Tell' was chosen because the system can display and speak any results of computation in any part of a user defined program. In STL, 'what you see is what it does'. For more detailed description of the language see [1] [2].

The language integrates three areas of computer application in the home and classroom environments, i.e., computation, database and communication. A future computer system will be used at home and school for managing communication, database, and computation through a single computer control language. There should be no need for learning different languages for different applications.

The integration is done with a uniform conceptual framework of *dataflow* plus *completion*. Completion refers to a problem in which the missing portions of an incomplete pattern must be filled in. A STL program is called a 'puzzle' because it defines a completion problem in the same way a jig-saw puzzle defines a completion problem. See Figure 1 for illustration of the concept of completion. A computation in STL is defined as completion by execution of operations, a communication as completion by dataflow, and a data-query as completion by pattern matching.

STL is implemented in C using the AZTEC C compiler on the Apple Macintosh™ personal computer with the AppleTalk™ local area network. The current version runs on a 512K Mac with an external disk drive. The size of the current object module is approximately 250KB.

2. PROGRAM STRUCTURE

A STL program (puzzle) consists of boxes connected by arrows. Loops and cycles may not exist in a STL puzzle. A box may be empty or may contain a data value, an icon which represents an operation, or another STL puzzle. Arrows allow data to flow from one box to another. An operation in a box will be executed when, and only when, all incoming values have arrived at the box. An empty box can be filled with a data object. Some empty boxes are used for communication with the environment of the puzzle. They are called the *base boxes* of the puzzle and are depicted by a thicker box frame. They correspond to formal parameters of a subroutine in traditional programming languages.

+ Show and Tell is a trademark of Computer Services Corporation (CSK).

A STL puzzle can be named by a user defined icon. Any Macpaint™ picture can be used as a puzzle name. When a box containing a user defined icon is evaluated by the STL interpreter, the puzzle (program) named by the icon will be evaluated, after the incoming data values are moved into the base boxes of the called puzzle. An icon is a subroutine name in STL. Recursive definition of a puzzle is also allowed. A recursive STL puzzle for the factorial function is given in Figure 2.

A STL puzzle is called *inconsistent* if two neighboring boxes contain conflicting objects. Otherwise it is *consistent*. When an arrow goes through an inconsistent box, data on the arrow does not reach the destination, ie., the inconsistency of a puzzle switches off the communication path provided by the arrow. There is no explicit Boolean data type in STL.

Iteration in STL is represented by a special kind of box called an *iteration box*. It contains another puzzle. The iteration box represents a dynamic array of copies of the puzzle contained inside. The puzzle will be copied repeatedly at execution time forming a dynamic array. The expansion of the array is terminated when the first inconsistent puzzle is produced.

Figure 3 illustrates a simple iteration with the Fibonacci sequence generation as an example. Figure 3.1 is a non iterative puzzle to compute the fourth Fibonacci number. Using an iteration box (the double-lined box), the same computation can be specified by the puzzle in Figure 3.2, where the iterative puzzle does not terminate, because no inconsistent puzzle will be generated by the interpreter during the dynamic copying process. The puzzle in Figure 3.3 terminates after 10 iterations producing the tenth Fibonacci number.

3. DATABASE APPLICATION

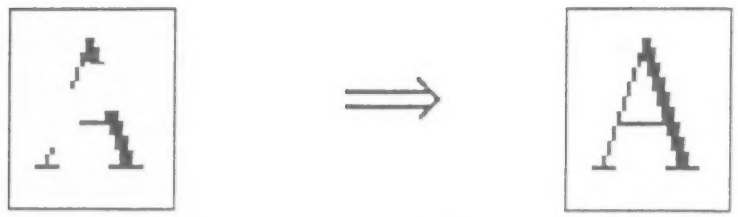
A set of data values is called a *solution* for a puzzle if the puzzle results in a consistent puzzle when the base boxes are filled in with the set of values. A set of solutions for a puzzle is called a *file*. Even though every puzzle can have a file, some may not have one. A STL file corresponds to a relation in a relational database, and the puzzle defines the *schema* of the file, i.e., the logical structure of the file. The base boxes of the puzzle define the *record structure* of the file and they are collectively represented by a *structure box* in the STL syntax.

Figure 4.1 is a STL puzzle defining a database schema for the Show and Tell personnel file. The puzzle currently displays one solution (record) contained in the file. Note that in STL, a data object can be a number, a text, or an image. Figure 4.2 illustrates how a file can be processed in STL. The puzzle outputs a profile (image) of a person with a given name as the input. The iteration box receives every record of the personnel file, represented by a file box containing the PERSONNEL icon, checks whether the name field matches with the input, and, if they match, transfers the value in the profile field to the next iteration. The structure box inside the iteration box decomposes a record into individual fields.

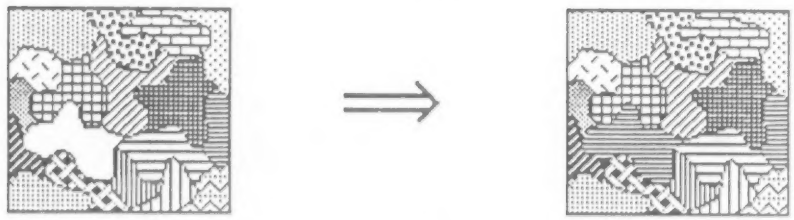
4. REFERENCES

[1] T. D. Kimura, J. W. Choi, and J. M. Mack. *A Visual Language for Keyboardless Programming*. Technical Report WUCS-86-6, Department of Computer Science, Washington University, St. Louis, March 1986.

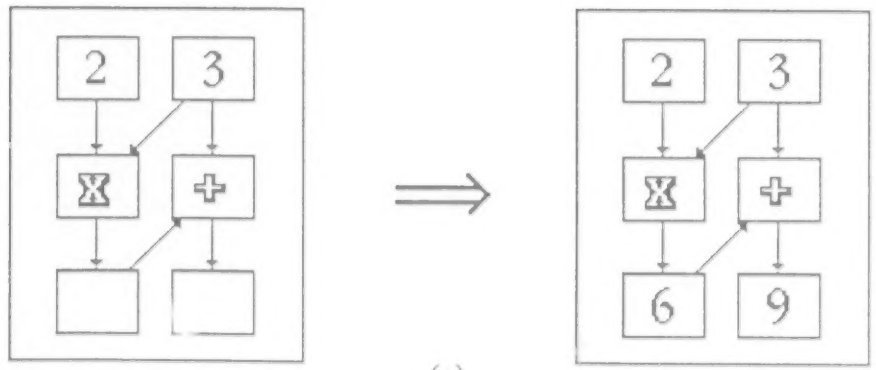
[2] P. McLain and T. D. Kimura, Show and Tell User's Manual, Technical Report WUCS-86-4, Department of Computer Science, Washington University, St. Louis, February 1986.



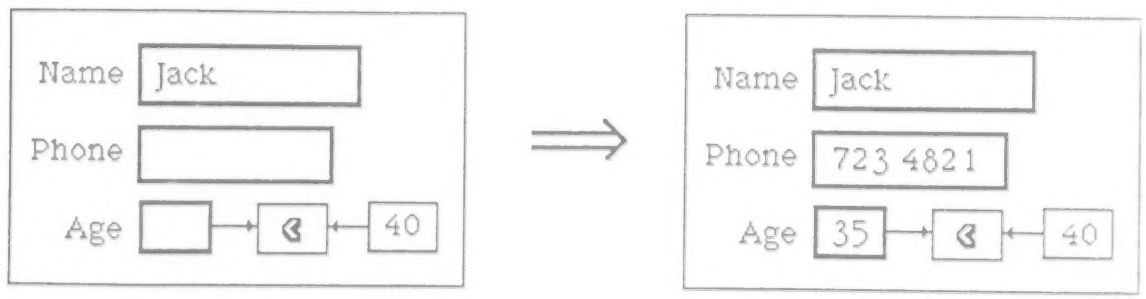
(a)



(b)



(c)



(d)

Figure 1: Completion Problems

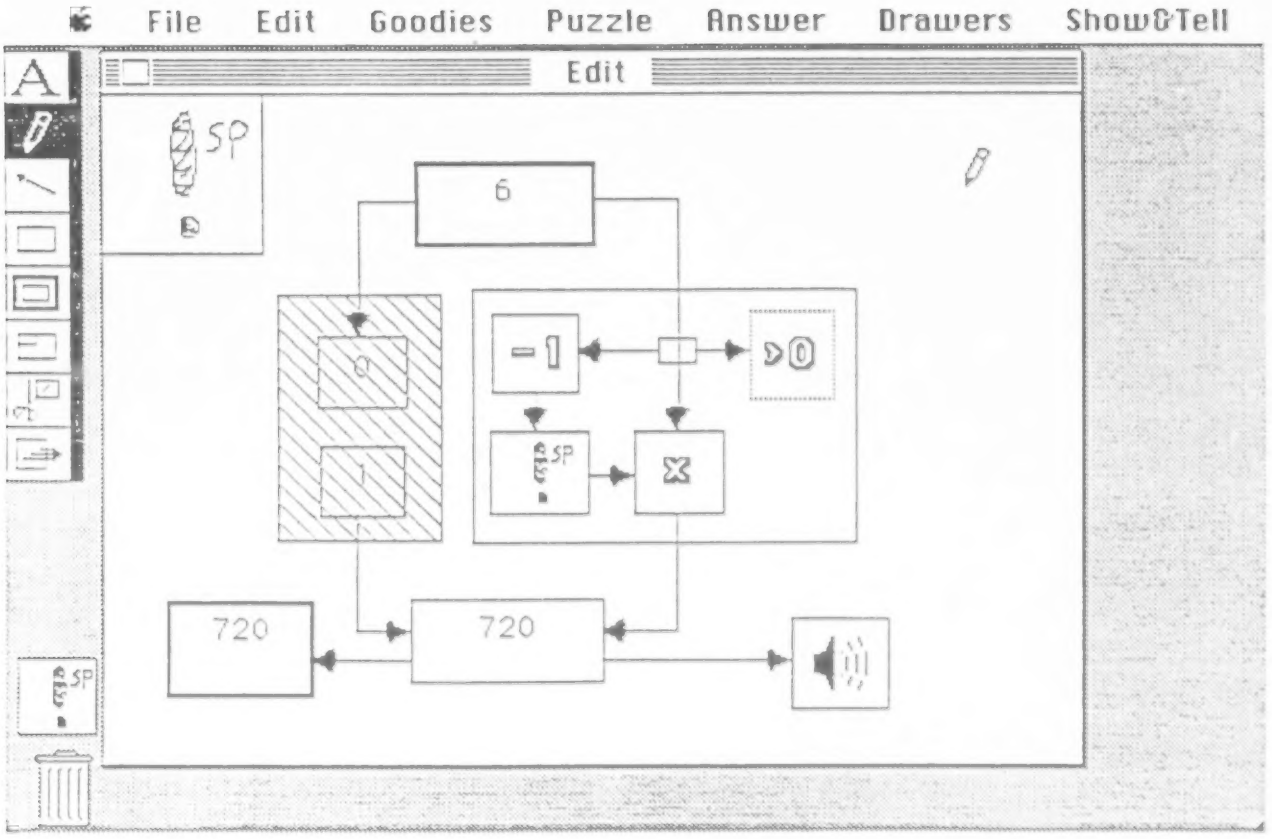


Figure 2: Factorial Function

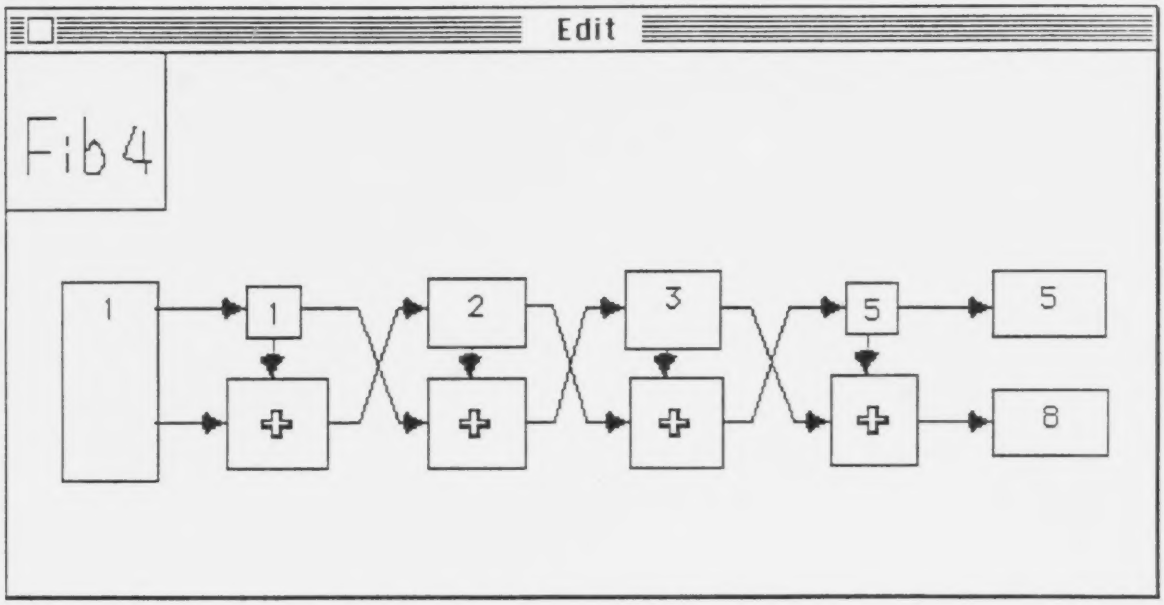


Figure 3.1: Fibonacci Number Program

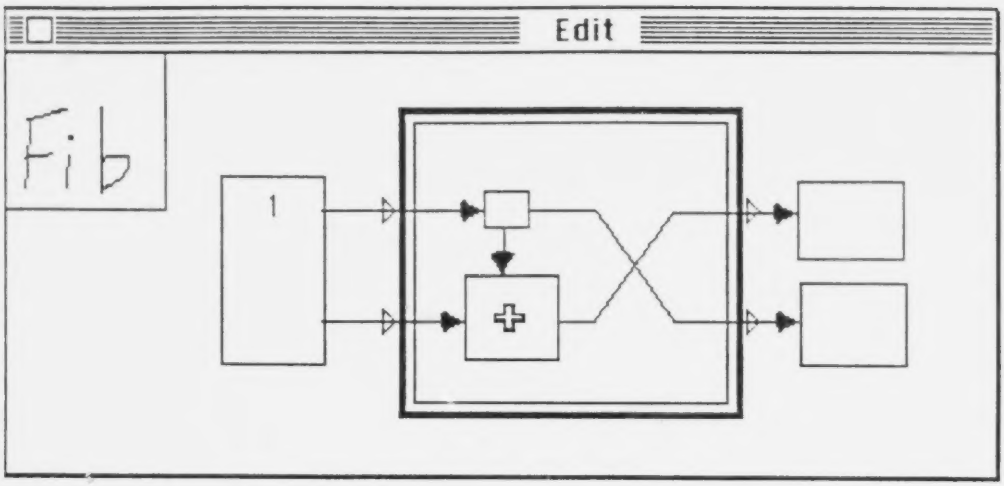


Figure 3.2: Infinite Iteration

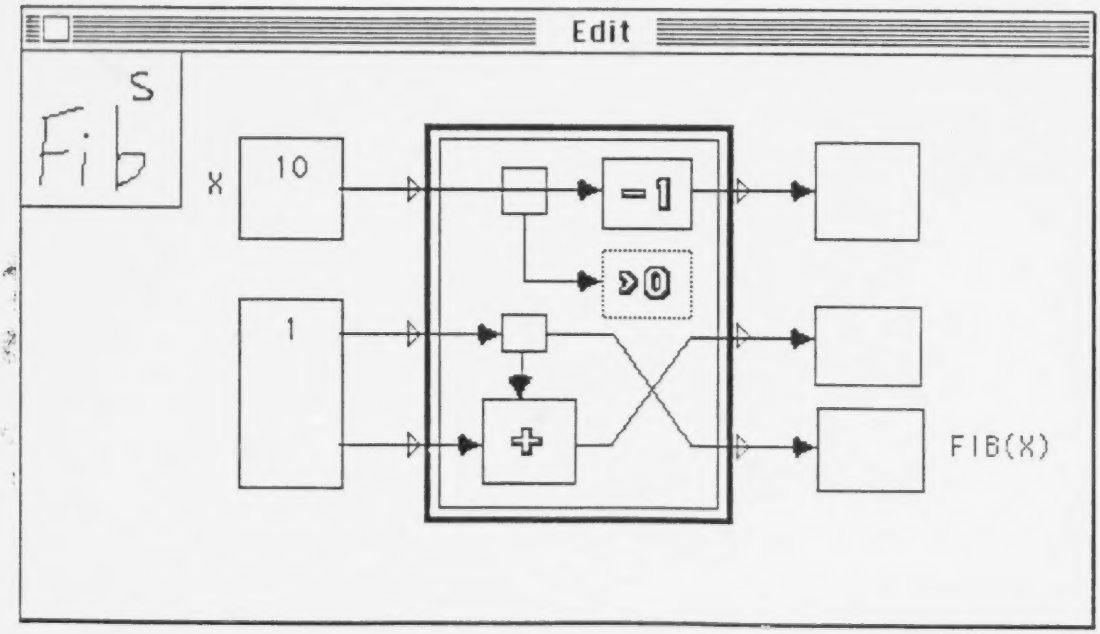


Figure 3.3: Terminating Iteration

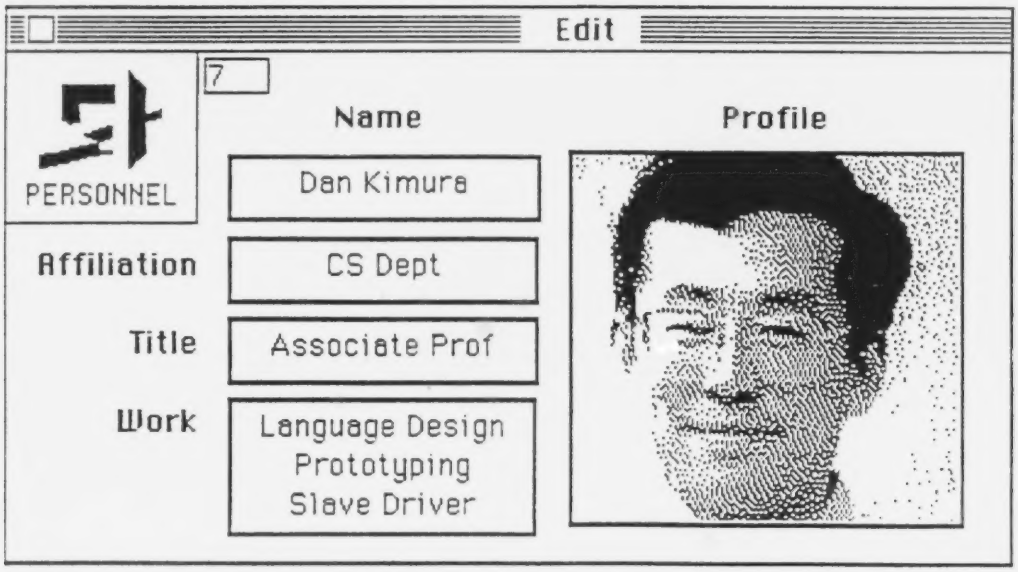


Figure 4.1: Database Schema

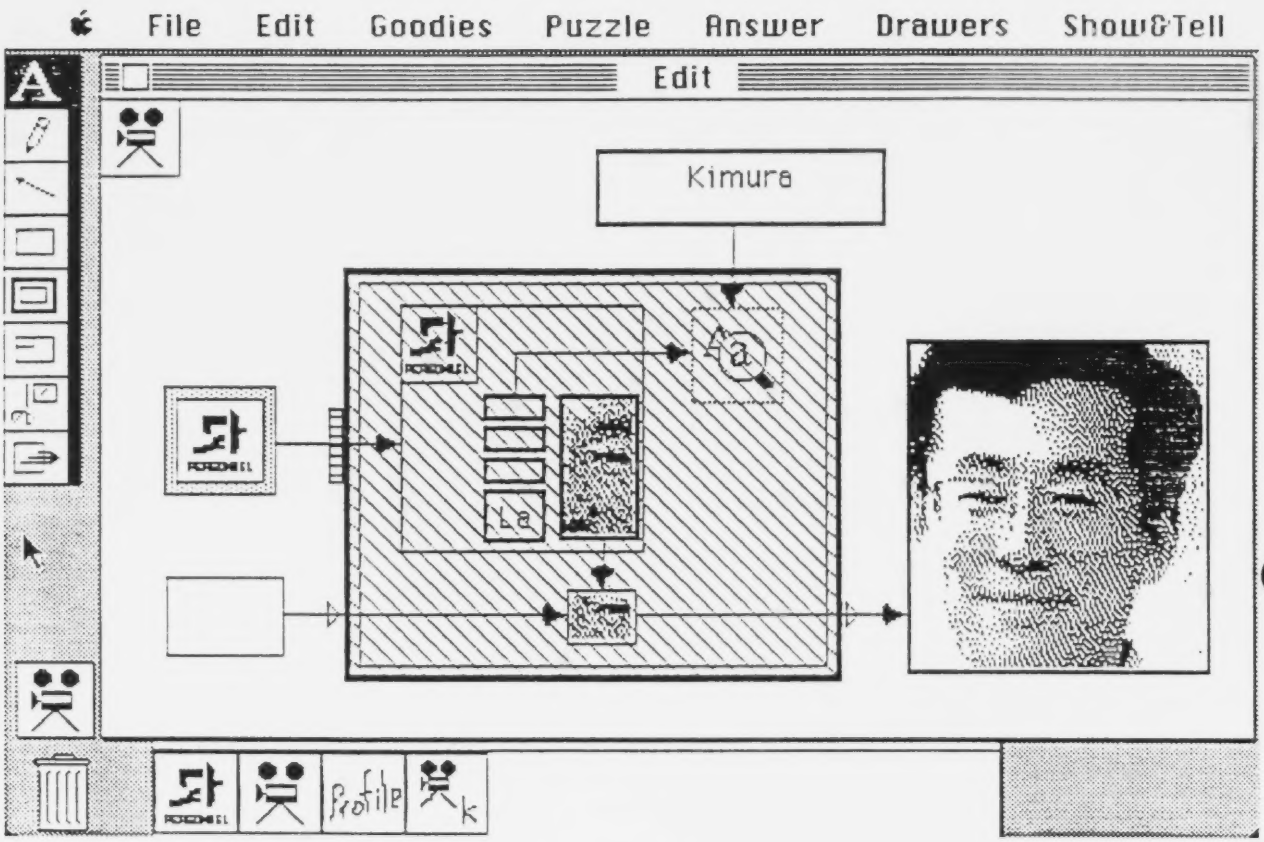


Figure 4.2: Data Query

THE GOSPEL ACCORDING TO ACKOFF, PRINCE, POLYA, RUBINSTEIN AND THE COMMITTEE

By: Peter Lykos, Illinois Institute of Technology, Chicago, IL

A research briefing is published every year (since 1982) by a committee formed by the National Academies of Science and of Engineering as well as the Institute of Medicine. The 1986 briefing included on its short list of priorities for the year problem-solving and decision-making (in addition to several topics in the physical and life sciences.

Increasing availability of data bases, management tools and support for simulation and modeling, including optimization, has broadened the power base of the problem solver and decision maker in some interesting ways. In a very real sense, access to such tools puts an even higher premium on human problem solving capability and, hence, on our ability to help humans perform better as problem solvers.

Thus, easier and easier access to more and more powerful such support tools forces us to learn more about how humans problem solve, that is, puts even more value on human factors.

Some persons consider that problem solving and decision making is the essence of the condition of being a human being - we do those things all of the time. Within the professions presumably professionals have gone through a program of study preparing them to be problem solvers within a certain range and within certain classes of problems. Where problem solving touches human beings there is a challenge of being competent in some related technology while at the same time being able to relate to the humans for whom solution of the problem is important.

There are two professional areas where the approaches to problem solving have been heavily skewed--in engineering toward the physical world and in management sciences toward the world of human interactions. They have come to overlap in that increasingly both are using computer-based methods and where increasingly the relevant factors involved in those problem areas come to include both physical world considerations and human attitudes, factors and approaches to problem solving.

In the summer and fall of 1974, I embarked on an odyssey on behalf of a major curriculum development project at IIT, E^3. The acronym E^3 stands for Education of Engineers through Experience. The basic design of the project was to develop a curriculum whereby an undergraduate could earn a bachelor's degree through self-study of basic concepts such as calculus, physics and related by working through modules (about the amount of material covered in about one week in a corresponding lecture/recitation course). In parallel with mastery of concepts, and discussions of social science-related subjects via reading and seminars, the students were organized into teams (of eight with two freshmen, two sophomores, etc.) to define and to find solutions for problems with a technical base, but whose solution would depend on human factors in terms of the context of the problem situation.

My charge was to bring the computer into E^3 (as one of some 60 of the IIT faculty drawn into the project on a part time basis). I spent the summer of 1974 travelling, reading, talking to people and designing what I would do in E^3 in response to the challenge. In the fall of 74, I worked with 11 E^3 students through the program I had designed for them.

The essence of the program was as follows:

(a) Each member of the team had a need to communicate with every other member of the team, with the two faculty advisors per team, with their peers in E^3 and with the rest of the world the relevant features of the technological problem, the technological solutions possible and the appropriateness of each given the circumstances. Thus, computer-based simulation and modeling became the foundation and the vehicle.

(b) The need to problem solve as a team required bringing out into the open just how groups go about problem solving as a group and how well prepared we are by our culture to do that.

(c) The advent of the computer made available problem solving tools which the participants needed to learn about to the point where they understood what the tools could do and could not do and how to use those tools.

In the process we learned that there is a discipline called Systems Science with journals and international conferences; that there is a simulation and modeling society with journals and semi-annual conferences, and that there exist not only computer software components which can be used to build models, but, in addition, simulations which are complete unto themselves with documentation such that several persons come together and role play within a dynamic and responsive model of some aspect of problem solving in our society.

The persons named in the title of this paper were called out as each had done work which contributed to what we did as follows:

Russell L. Ackoff spent his entire professional career as a problem-solver for hire. At the end of his career, he felt the need to set down in writing what he had learned. He titled his book, The Art of Problem Solving; (Wiley, 1978). His thesis was that good managers are characterized by the five Cs; Competency, Communicativeness, Concern, Courage and Creativity. He focused on Creativity. The principal part of his argument was that creativity is something we are all born with and exercise as children, BUT that we are taught to conform and are acculturated with habits which stifle creativity. He reached back to the 80 years of the ancient Greek Civilization to display the philosophy of the ancient Greeks and the four pursuits of man:

Scientific.....Pursuit of truth
Political-Econ.....Power and plenty
Ethical-Moral.....Goodness and virtue
Aesthetic.....Beauty

Ackoff got into the role of aesthetics in problem solving, the notion of setting forth an idealization of the situation and of working with an abstraction initially. He emphasized regeneration of our creativity and through examples sought to display what he felt was important on the part of the individual's attitude. Ackoff's book tends to be found in graduate programs in management science.

George M. Prince eventually formed a company based on using the synectics method. He developed an extensive clientele for he had learned that within organizations there is generally a less than optimum attitude and corresponding practice on the part of problem solving teams. His focus became the company meeting and what might be done to make that a more effective means of problem solving. He also brought out that our culture provides us as the major lessons in living, (home, school, play, work) to COMPETE, to LOOK OUT FOR NUMERO UNO, and to do that POLITELY, IF POSSIBLE. As a direct consequence, our behavior in a group problem solving situation tends to be destructive HABITUALLY, UNCONSCIOUSLY and UNINTENTIONAL (probably). Prince emphasizes interpersonal relationships and how they affect the ability of a group to problem solve.

By way of a demonstration of those tendencies, and also to bring out an important feature of group approaches to problem solving, we demonstrated the Desert Survival Game. A company named Human Synergistics put together a game where individuals in a large group are presented with a scenario. They are to imagine they are part of a small group in an airplane crossing a large desert. The plane crashes and the survivors are given a list of objects which they were able to salvage from the wreckage. First each individual is asked to list in priority of importance to survival the objects salvaged. Then the large group of persons is subdivided into several groups of six or seven each and each of those small groups is asked to repeat the process WORKING AS A GROUP. The two sets of results are evaluated using the opinions of several experts in desert survival as a basis for the 'correct' ordering of the objects. The amazing result established empirically by the members of the group themselves is that small groups can and usually do do better than the individuals!

George Polya , in 1945, outlined in his book, How to Solve It, four phases of problem solving. Because Polya was a mathematician, his examples were drawn from that field and so are inaccessible to those who fear mathematics. Polya first laid out the notion of a heuristic approach to problem solving (heuristics is the essence of so-called expert systems). Even though the problem domain is mathematics, he focused on the problem solver! He also brought out the fact that we have a conscious and a subconscious and that after we have studied the problem and all of the evidence intensively, we should ruminate--take a walk, sleep, turn the problem over to our subconscious. Polya emphasized proverbs, taking the position that proverbs represent a distillation of human wisdom over the ages, our heritage! And he developed a set of guidelines, showing their origin in proverbs, for approaches to problem solving. Algorithmic thinking has been related to Polya's approach.

More recently, and reflecting easy access to computers, **Moshe Rubinstein** picked up on mathematical modeling with attention to the fact that most problems have a human dimension and evolved at UCLA a course and subsequently a book, Patterns of Problem Solving (Prentice Hall, 1975). His book is a grand tour of the important ways we can go about problem solving using tools now easily accessible to us via computers. He distinguishes between discrete state and continuous modeling, as well as hybrids. Indeed, in E^3 we were able to use Rubinstein's book as a basis for a weekly seminar/group meeting (at the rate of one chapter per week) AND to complement that book with a computer-based modeling environment, GASP IV by Pritzker at Purdue. It was at this juncture that we attempted to bridge what we had learned from the human-focused management science methods and tools, and what engineers with their emphasis on physics and mathematics traditionally learned to do.

Following that E^3 experience, I was called to organize and to conduct a day and a half intensive workshop for 85 persons who were the staff running a production line in a major high tech industry. The formal education of those persons ranged from high school diplomas only to PhDs in some branch of engineering.

Our meeting in 1982 on creative approaches to problem solving incorporated much of what we had learned in E^3. In the evaluation of that workshop, the strong response to the dimensional analysis component (a small part of the workshop) I included, prompts me to summarize that portion here. In dealing with some dynamic process, there is a set of independent variables whose values at any point in time determine what the dependent variables, or outcomes, take as values. The function whereby the relationship models what is happening may be unknown in detail, but whatever form it takes the units must be consistent with what is observed. Thus, in a physical systems one may be dealing with Mass (symbol, M); Length, L; Time, T; Velocity, LT^{-1}; Acceleration, LT^{-2}; Density, ML^{-3}; Force, MLT^{-2} and Work, ML^2T^{-2}.

If one is studying the flow rate, Q (L^3T^{-1}) of a liquid with viscosity coefficient eta (more later) through a tube of length (el) and radius R (L) under a pressure differential P (pressure is force per unit area or MLT^{-2} times L^{-2} equals $ML^{-1}T^{-2}$) then one can assume a model where:

$$Q \text{ is proportional to } p^a (el)^b (eta)^c R^d$$

The viscosity is defined as force per unit are per unit velocity gradient or:

$$(MLT^{-2})(L^{-2})(LT^{-1} \text{ times } L^{-1})^{-1} = ML^{-1}T^{-1} \text{ for the units of eta.}$$

Thus one has the condition:

$$L^3T^{-1} = (ML^{-1}T^{-2})^a L^b (ML^{-1}T^{-1})^c L^d .$$

Therefore one has the conditions (the exponent of each unit must be the same on both sides of the equation!):

for M, $0 = a + c$

L, $3 = -a + b - c + d$

T, $-1 = -2a - c$

Thus already one has reduced a situation with four unknowns to just one unknown.

That is just one specific instance where one is considering explicitly the constraints that come with the physical system one is modeling.

A related and more general situation obtains when one is seeking the optimum one of a large number of possible solutions to a problem (an aspect of the discipline, Operations Research). Here a set of simple linear equations which constitute a model of the problem are complemented by a set of inequations which express the constraints. (The method is called linear programming - which has nothing to do with programming a computer).

Finally, moving on from Rubinstein's co-mingling of human factors and human values together with mathematical modeling of physical phenomena, we come to the last word in the title, the Committee. The publication, Research Briefings 1986, is available from the National Academy of Sciences, 2101 Constitution Ave., NW, Washington, DC, 20418. The NAS identifies what its members consider to be research areas RIPE FOR SIGNIFICANT BREAK THROUGHS by way of guidance to the cognizant congressional committees which support basic research. For the first time (since 1982) there appeared on the (short) list of topics, Report of the Research Briefing Panel on Decision Making and Problem Solving. The report,which is some 16 pages long, lays out what the state of our knowledge is about those subjects and suggests areas which are pregnant with possibilities.

The focus of that report is on prescriptive theories of choice and how those are complemented by empirical research that shows how people actually go about making decisions. That essay is mentioned here, not so much because of its content which is rather abstract, but rather by way of bringing out that the experts in the field feel strongly that the time has come to focus on decision making and problem solving as a subject deserving of attention in its own right.

The computer as an extension of the human mind in problem solving has many dimensions and enormous power both innate and because of its easy access. The rate at which that power is increasing far outstrips our ability to use it. Part of the problem stems from the fact that we do not stress the human phenomenon of problem solving and of what is known about that process in our academic programs.

That is the challenge.

GIFTED EDUCATION USING MICROCOMPUTERS

By: John C. McDonald, Grayslake Community High School, District #127, Grayslake, IL

HISTORY: Grayslake Community High School District #127 is a one school district with an enrollment of approximately 850 students. During the 1983-84 school year, the district initiated a Gifted Education Program for students of English, math and science. The emphasis of the program was to utilize microcomputers to allow students to explore topics not in the regular curriculum or to use microcomputers as an alternative mode of instruction. In the current school year the program has expanded to include students of English, math, science, computer science and journalism.

SELECTION: Students are selected for the Gifted Education Program based on four (4) criteria. The criteria for selection are one (1) self nomination, two (2) teacher nomination, three (3) intelligence test score and four (4) aptitude test score. We do not use specific cut-off values on the tests, but rather assign a value to each of the tests used on a scale from 1 to 7 (7 high). For example, if all Freshman students in school have a median IQ of 100, then any student below the median is assigned a score of 1. Students with IQ scores from 100 to 110, a score of 2, 111 - 120, a score of 3, etc. In the final selection, those students who have indicated an interest in the program, and who have been nominated by a teacher are chosen from the highest score attained on the IQ test and aptitude test until we arrive at our maximum state funded enrollment of 35-40 students.

SCHEDULING: Students with regular study halls meet three (3) periods per week (50 minute periods). Students who have only a "homeroom" type study hall meet five (5) times per week (20 minute periods). A student who has good attendance in the program and meets the course objectives and achievement expectations obtains one (1) semester credit at the end of the school year. This credit is weighted as a regular level semester credit and is included in the student's grade point average.

ASSIGNMENTS: Applicable software, programming projects and use of word processing is assigned to the Gifted Education Program students. The students work with teachers of their current classes in their discipline area or areas. In addition to working with their own teachers, the students are expected to make contact with a liaison teacher for each of the areas in the program. Students are expected to share in the responsibility for determining their software, programming and word processing assignments.

OBJECTIVES:

* To utilize CAI (Computer Assisted Instruction) and computer programming to provide optional and additional modes of learning interdisciplinary content.

* To allow students to create computer programs related to course objectives in accelerated curriculum.

* To allow students to utilize their interest in computers in meeting interdisciplinary learning objectives.

* To explore current topics in English, math, science, computer science and journalism not offered in the regular curriculum.

EVALUATION:

* Students who are assigned programming will write no less than three (3) tutorial programs of a "user friendly" nature appropriate for use as CAI related to the goals of accelerated courses currently enrolled in.

* Students in all areas will write no less than three (3) reports explaining the uses of particular software packages as related to the goals of accelerated courses currently enrolled in. These reports will include an evaluation of the software from the user's point of view. They should also include, but not be limited to, suggestions for the implementation of the software packages into regular classes.

* Students in all areas will submit a weekly progress report to the program coordinator. The report will include any rough drafts of projects upon which the student is working. This includes preliminary listings of programs being written. Any problems the student is encountering may be put in writing on these weekly reports, allowing the program coordinator the opportunity to respond.

* Assignment of grades will be based on the assessment of the student's work by the program coordinator in conjunction with the content area contact teacher.

EQUIPMENT: The students enrolled in the Gifted Education Program at Grayslake Community High School have access to microcomputers during regular school hours only. There are two (2) computer labs, one with 14 Franklin Ace 1200 computers and one with 17 TRS-80 Model III computers. Also available to the students are 2 Apple IIe computers, 2 TRS-80 Model III computers, 2 Franklin Ace 1000 computers, and 5 Franklin Ace 1200 computers in outlying areas of the school building.

ACADEMIC AREAS:

ENGLISH: Students who choose this area of study are more often than not exposed to less software than students in other areas. We do have available to them software that can help them learn speed reading techniques, write a character sketch, learn antonyms and synonyms and analogies, and analyze novels. Usually the students in this phase of the program are assigned creative writing assignments which require they use a word processor for their rough and final drafts. The papers they write can run the gamut from poetry to novelette. In addition to the creative writing assignments, the students read from a monthly magazine and do a variety of assignments from those indicated in the magazine.

MATH: The students enrolled in this area are expected to do programming assignments closely related to their regular math class curriculum. We view these programs as an enrichment rather than an acceleration, although there are times when circumstances dictate acceleration as well. If students come to the gifted program with no programming experience, they are expected to learn enough BASIC to perform their assignments. We have available to these students software that will enable them to add graphics to their programs via the software or via a graphics tablet. Math software made available to these

students includes several plotting programs, ACT and SAT math programs, word problem tutorial programs, probability programs, arcade type math programs and programs to help build problem solving skills. These students also study programs found in the Mathematics Teacher published by NCTM.

SCIENCE: Over the years we have found that this area of study has the lowest enrollment. Perhaps this is due to the fact that we do not have "tracking" throughout our regular science curriculum. The students that do enroll in this phase are expected, much as the math students, to be programmers or become programmers. They are assigned projects to help them with their regular classes in the biological sciences, chemistry and physics. Some students branch out into astronomy and space studies. We try to be very flexible in our willingness to approve student suggested assignments, to give them an opportunity to be creative. Software available to the science students allows them to study ecology via simulations, chemical bonding via animated graphics, and biological and physics experiments via interfacing of light and heat sensors. These students also study programs found in popular magazines pertaining to their field of study. Often they use the graphics tablet and software to incorporate graphics into their programs.

COMPUTER SCIENCE:
We have found that often we have students who are interested in computer programming after taking our limited courses available in the regular curriculum. These students are qualified for the Gifted Education Program whether they are currently enrolled in computer science classes or not. They are assigned programming tasks commensurate with their abilities and interests. Often these students learn assembly language or Pascal (not offered in the regular curriculum) on an individual basis. In addition, we allow them great flexibility in the use of software and interfacing hardware used in the other areas of study. This is done to allow them to see the computer as a tool for programming and as a tool for the non-programming user.

JOURNALISM: In the past two years of the program we have developed a journalism field of study. These students fall into essentially two categories. First, we find those students who are enrolled in regular journalism class who work beyond the course requirements. Second, we find those students who have either finished the journalism curriculum or who can not fit regular journalism into their schedule. In both cases, these students work closely with the journalism instructor. At present the students in this group merely use the microcomputers to write their articles with a word processor. We are not yet to the point of having the typesetting done directly from our computers or computer disks. Actually, we were running such a system in the past year, but technical problems prevented us from continuing this year. The students and sponsor are now in the process of studying the concept of desktop publishing for possible use in the near future. Students assigned to the business end of the school newspaper do use database management programs to keep the financial records in order.

WHAT KINDS OF PROBLEM SOLVING SKILLS DO KIDS LEARN WHEN THEY LEARN PROGRAMMING?

By: Diane McGrath, University of Illinois, Urbana-Champaign, IL

For some reason which has always puzzled me, teachers are expected to teach subject A and kids are expected to learn both subject A and some other secret thing, B. For instance, when you learn math you are supposed to learn problem solving. When you learn Latin you are supposed to learn to think logically. And so on. It's like a great conspiracy: don't tell the kids what they are supposed to learn. Don't even tell some of the teachers. Yet the state legislatures think that students are learning problem solving or logical thinking as they are being taught other disciplines. If they don't learn it, the teaching must be poor and/or the kids weren't that bright anyway. If they do learn it, then those are smart kids and they should get ahead. Not too many people examine the evidence for the claim itself.

One of the most popular forms of teach A - learn B is the current rage to teach problem solving by teaching programming. This version is inspired by claims that kids learn to solve problems by learning Logo, and further that some of their problem solving abilities have hitherto been delayed by not having adequate tools to think with (Papert, 1980). I have to confess that I find it a reasonable and interesting possibility. Programming after all is the process of getting a computer to solve a problem for you. Various algorithms may be used over and over regardless of the precise content, and programming is writing a structure that is not necessarily bound to a particular content.

I'd like to share with you the results of the 18 or so fairly decent pieces of research on the question, and then suggest 4 factors which I believe are crucial in making it most likely that students will learn problem solving while learning computing.

WHAT MIGHT TRANSFER?

Before we look at the studies, however, let's first look at the things that are most likely to transfer from programming to some non-programming situation:

At the most abstract level we might say that the problem solving and programming processes both involve these four phases (see Polya, 1945; Dalbey & Linn, 1986; Pea & Kurland, 1984):

1. Comprehending the problem: givens, goals, obstacles
2. Planning a solution: flowchart or verbal outline
3. Implementing a solution: writing the code
4. Debugging

At the most concrete level, we might expect that kids could learn to use well some elementary skills that they practice a lot in programming (Shneiderman, 1985):

1. Making comparisons: does A=B? is C<F?
2. Examples of concept ("words" is a string, D is a variable, 32.4 is a number)
3. Idea of repeating a sequence of events (loop)
4. The conditional: IF----THEN----ELSE----

At various levels, some strategies that one learns in programming might be applicable to other classes of problems (see, e.g., Linn, 1985), for example:

1. Binary searching
2. Modular design, with re-usable procedures
3. Breaking down larger problems into manageable problems
4. Using an old procedure, slightly modified (parameterized) in a new context, or even in the old one for a new effect
5. Recognizing that the structure of the new situation has some essential features in common with that of the old, and thus being able to solve the new problem by analogy to the old. This may not be so obvious as some other things, an perhaps we don't teach it. But programming looks to me like a good place to try to teach it.

WHAT DOES THE RESEARCH SHOW?

The clearest way to see the research findings is to to view the following three tables, which contain lists of those skills and tasks which have and have not been found to show transfer at three different age levels.

Elementary

Of the 8 studies on K-5 students, 5 used Logo only, 1 used BASIC only, and 2 compared BASIC and Logo. Half of these found that there was transfer from programming to some other measure: half found none (See Table 1). What conclusions can we draw from these studies?

1. You can't catapult kids into the stage of formal operations by a few hours of Logo (6 hours to 60 hours).

2. Certain reasoning and planning abilities are not improved either.

3. However, the nitty gritty things that kids practice at that age in a programming environment seem to be improved: classifying, comparing, deciding what to do IF something happens or IF it doesn't.

Middle School

See Table 2 for a summary of what did and did not transfer at the middle school/junior high and high school age. Two studies in middle school are worth discussing in more detail, since they start to get at some of the complexity of the issue.

Dvarskas (1983) found that some programming students improved on math word problems. She looked at concrete operational students and transitional operational students in middle school, in both BASIC and Logo conditions, and found that both computer languages helped tremendously (as compared to the control group) for the concrete students! In other words, perhaps Papert was on the right track with regard to this age group. The transitional kids at this age were already as advanced as they were going to be; the slower developing kids moved faster and farther.

Kurshan et al, (1983) tested students on the problem solving subtest of the SRA achievement test (the math portion of the series, probably math word problems) and found that programming students improved on this test more than did the control group. Note, and I will come back to this, that this was true only for those subjects who had had

previous programming experience - in other words, they were on their second programming language.

There is only 1 high school study. This in itself is an interesting fact in light of the pressure from the various school reports to require a programming course of all high school students so that everyone may learn problem solving. Cheshire's research showed no developmental gains as measured by the DCAT.

Conclusions: Middle School and High School

1. Developmental level itself is not changed dramatically, but kids in the concrete level seem to be brought along in the area of math word problems.

2. Certain other skills at a fairly low level seem to be improved: finding missing pieces, matching patterns, predicting picture from directions, writing directions, creative design - things that are above the comparing and classifying level, but not yet at the cognitive level many researchers and public servants had hoped for.

In addition, several studies show that at middle school and junior high school level there is transfer from one programming language to another, a rather unsurprising finding (Mandinach, 1986; Dalbey & Linn, 1986).

College

There have been four college studies. These are summarized in Table 3. Generally they have found evidence for improvement in math word problems, but not for strategies for solving them.

1. Skill at math word problems and some of the components of these problems are enhanced by programming for a period of from 20 hours to a semester or more.

2. Strategies for efficiently solving problems that border on math word problems are not improved by 20 hours of programming experience.

A CLOSER LOOK

It seems to me there are several factors at work which need a close examination.

1. **Time**: The most obvious factor that might affect how well you learn the secret skills is that of how much time you spend practicing the original skills from which they are expected to transfer. In fact, kids in these studies rarely learned programming well enough, much less well enough, to abstract anything to take with them to a problem solving task. In the 13 pre-college studies, 11 of them taught programming from 6 hours to 60 hours, and an additional 2 taught it for nearly a year. It would be very surprising to me to find a skill that is not well learned in the first place helps you in another arena.

2. **Explicitness**: As I have been implying right from the beginning of this article, why keep it a secret? If you are expecting a youngster to learn a skill, why not tell her? Say out loud what the skill is and show how it might be used in several different arenas. None of the studies mentioned taught problem solving explicitly, but many of them mentioned that it should be taught explicitly.

3. **Near and Far Transfer** : We may look for transfer of skills in areas that are very similar and have many common elements to programming, or in areas that are quite far afield. These studies covered the range. Studies which found transfer found it in: another programming language (very near transfer); math word problems (near transfer, I'd say, since that is another formal system and one which manipulates variables in a similar way); and the ability to use concepts classification and conditional rule and a number of other skills that seem to be very closely related to skills actually practiced in programming at the level at which it was taught.

The research on transfer suggests that if you want far transfer - if you want a skill to be able to be used in widely different contests, you must give the person extensive and varied practice on that skill (Perkins, 1985). Most of these students don't even have extensive practice on a skill, much less varied practice. Varied practice would mean giving the students problems that use the same skill, but that are not programming problems. It would take a lot of work on the part of the teacher to think up such problems, but it seems to me to be crucial if we are trying to teach kids to be better problem solvers in a society that sorely needs some of its problems solved.

4. **Analogical Transfer**: One of the most promising lines of research for our purposes may be something which is coming out of cognitive science, and that is the study of analogical problem solving: the solving of one problem by making an analogy to another. Think about it - in order to solve a problem in a new area by using a skill or concept or algorithm used in programming, you must either invent the solution from scratch or make an analogy to the programming situation - you must see that situation B is similar in some crucial way to situation A found in programming. There are 2 ways this might be done:

a. Direct Mapping: of the structure of A onto the structure of B - you may simply notice that they are similar, because of some feature(s) that they have in common. Then you say hmm, I wonder if the same solution might work here. In the literature on this subject, people make spontaneous analogies like this in perhaps 10-20% of the cases in which such an analogy is possible (Gick & Holyoak, 1983).

b. Schema Induction: If you have the opportunity to see 2 or more problems with a similar structure and similar solution, then you have the possibility of abstracting something more general about this situation - seeing that there is a structure and solution which go together. This increases to over 90% the chance that when you encounter a third problem of this type, you will be able to more readily see that it is of the type with that structure. This figure increases to still further when you add explicit explanations (Gick & Holyoak, 1983).

The way that this research applies to programming, I think, is that you cannot expect to see transfer of a skill learned in programming if that skill is not well enough learned to see the abstract structure of the problem and solution. The abstract schema makes it far easier to solve new problems. But what does it take to get the abstract structure in the first place? It takes at least 2 instances and some explicit instruction. Two instances of what? I suggest that it takes 2 instances of a programming language. Not 2 instances of a problem involving converting hours to minutes or quarters to pennies. In other words, students in their second programming language are much more likely to see that they have learned some things about loops, algorithms, debugging, and the like than are students in their first programming language, where these concepts are so tied to the first language commands and whether or not they are being used properly. And this goes for teachers as well. I might add: how can we understand these abstract ideas if they are tied to the concrete instance of a single programming language? We must see the ideas played out in several programming languages too.

CONCLUSIONS: Suggestions For Teachers of Programming

1. Think about the age group you are teaching in terms of what this research has shown. In elementary school, emphasize the elementary concepts like classifying, comparing, repeating, and getting a good grasp of branching through the conditional. Kids understand PART of the IF..THEN.. when it's in the form: IF you clean your room THEN you may go out and play. You may have noticed that they don't so readily grasp what happens ELSE. Programming is a good environment in which to teach different forms of the conditional rule.

In middle school, emphasize planning, predicting, following directions, giving directions, fixing procedures that don't work

In high school, emphasize procedures that may be used again in different way, debugging methods, planning, algorithms, etc.

2. Give enough time. It takes about 50,000 chunks of information to become an expert (Simon, 1980). Kids will not become experts. But they must have enough time at the computer to master the commands and procedures they are learning. This means that 1/2 hour once a week won't do it. Nor will a 3-week unit on programming.

3. Teach them explicitly what you want them to know. Don't just hope they will understand that a procedure is to be used again or that a problem should be first broken down into parts. Discuss it with them and show them how this can be done in a number of examples. Tell them the principle as well.

4. Give extensive and varied practice. Use non-programming examples and homework problems that use the general skills you hope to build.

5. Teach a second programming language as well. This probably doesn't apply as much to elementary school teachers, although I could imagine a curriculum in which youngsters learn both BASIC and Logo in elementary and middle school, and the teacher of the second language could make the similarities explicit.

But certainly in high school, when a second programming language is available, the teacher of that language should be especially concerned that those students, the ones who have a high potential for seeing the connections among the various types of problems, should be given the explicit training, and extensive and varied practice, that promotes the widest application of the skills.

Finally, I do not believe or mean to imply that all problem solving is best done in a top down fashion. It somewhat horrifies me to read programming proponents who believe that this structured approach will be the best way to tackle any problem. I see programming as only one arena in which to teach problem solving skills, and then only one variety of problem solving skill. There is some evidence that this approach is seriously limiting (Hayes-Roth & Hayes-Roth, 1979).

I would hope that problem solving would be taught in English classes, journalism classes, social studies classes, and so on. Furthermore, I would expect these same principles to apply: Think about the age group and those problem solving skills they might be reasonably expected to learn. Give them enough time working on problem solving. Don't

have a hidden agenda - teach the skills you want them to learn explicitly. Give them extensive and varied practice. Make sure they have enough examples to abstract some generalizations about when the method is appropriate to apply and when it is not, so analogical problem solving will be enhanced.

All education should be about problem solving.

REFERENCES

Chesire, F.D. The effect of learning computer programming skills on developing cognitive abilities. Unpublished Doctoral Dissertation, Arizona State University, 1981.

Clements & Gullo, Effects of computer programming on young children's cognition. J. Ed. Psych., 1984, 76:1051 - 1058.

Dalbey & Linn, Cognitive consequences of programming: augmentations to BASIC instruction. J. Ed. Comp. Res., 1986, 2:75-93.

Degelman, D., J.U. Free, M. Scarlato, J.M. Blackburn & T. Golden. Concept learning in preschool children: Effects of a short-term Logo experience. J. Ed. Comp. Res., 1986, 2: 199-205.

Dvarskas, D.P. The effects of introductory programming lessons on the learners' ability to analyze mathematical word problems. Unpublished Doctoral Dissertation, The University of Connecticut, 1983.

Ehrlich, K., V. Abbott, W. Salter & E. Soloway. Issues and problems in studying transfer effects of programming. Symposium: Developmental studies of computer programming skills, presented at Annual Meeting of the AERA, New Orleans, April, 1984. Available as Tech. Report #29, Bank Street College Center for Children and Technology.

Gaffney, C.R. Computer languages: Tools for problem-solving? Unpublished Doctoral Dissertation, The University of Texas at Arlington, May, 1984.

Gick, M.L. & Holyoak, K. Schema induction and analogical transfer. Cog. Psych., 1983, 15: 1-39.

Gorman, H., Jr. & Bourne, L.E., Jr. Learning to think by learning LOGO: Rule learning in third-grade computer programmers. Bul. Psychon. Soc., 1983, 21(3): 165-167.

Hayes-Roth, B. & Hayes-Roth, F. A cognitive model of planning. Cog. Sci., 1979, 3: 275-310.

Horton, J. & Ryba, K. Assessing learning with Logo: A pilot study. The Computing Teacher, Aug/Sept, 1986, 14: 24-28.

Kurshan, B., J. Williams & N. Healy. Does the use of microcomputers in junior high school increase problem solving skills. Proceedings of NECC/5, The Computer Society Press, 1983, 311-315.

Linn, M.C. The cognitive consequences of programming instruction in classrooms. Ed. Researcher, 1985, v. 14(5): 14-16ff.

Mandinach, E.B. Aspects of programming courses that foster problem solving. Paper presented at the Annual Meeting of the AERA, San Francisco, 1986.

Mayer, R.E., J.L. Dyck & W. Vilberg. Cognitive consequences of learning BASIC computer programming. Paper presented at the Annual Meeting of the AERA, San Francisco, April, 1986.

Milojkovic, J.D. Children learning computer programming: Cognitive and motivational consequences. Unpublished Doctoral Dissertation, Stanford, 1983.

Papert, S. Mindstorms. Basic Books, New York, 1980.

Pea, R. Logo programming and problem solving. Tech. Report #12, Center for Children and Technology, Bank Street College of Education, 1983.

Pea & Kurland, On the cognitive and educational benefits of teaching children programming: A critical look. New Ideas in Psych., 1984, 2: 137-168.

Pea, R.D., Kurland, D.M. & Hawkins, J. LOGO and the development of thinking skills. In Chen, M. & Paisley, W. (eds.), Children and Microcomputers. Sage Publications, Beverly Hills, 1985, 1985, 193-212.

Perkins, D.N. The fingertip effect: How information-processing technology shapes thinking. Ed. Researcher, 1985, 14(7): 11-17.

Polya, G. How to Solve it. Princeton University Press, Princeton, N.J., 1945.

Rose, N.S. The effects of learning computer programming on the general problem-solving abilities of fifth grade students. Unpublished Doctoral Dissertation, North Texas State University, 1983.

Rucinski, T.T. The effect of computer programming on the problem solving strategies of preservice teachers. Unpublished Doctoral Dissertation, University of Illinois at Urbana-Champaign, 1986.

Shaw, D.G. The effects of learning to program a computer in BASIC or Logo on the problem solving abilities of fifth grade students. Unpublished Doctoral Dissertation, North Texas State University, 1984.

Shneiderman, B. When children learn programming. The Computing Teacher, February, 1985, 12: 14-17.

Simon, H.A. Problem solving and education. In Problem Solving and Education: Issues in Teaching and Research , D.T. Tuma & R. Reig (eds.). Erlbaum, Hillsdale, N.J., 1980.

Taitt, N.P. The effect of computer programming instruction on the problem solving ability of pre-service elementary teachers. Unpublished Doctoral Dissertation, University of Illinois at Urbana-Champaign, 1985.

TABLE 1. ELEMENTARY SCHOOL

Skills that have and have not been found to transfer from programming to non-programming environments. (K-5).

What Transfers?

Reflectivity	(think before you act)
Divergent thinking	(number of ideas; original ideas)
Awareness of not understanding	
Creativity	
Ability to describe directions to another	
Classifying	(simply and conjunctive concepts)
Conditional rule	(IF.... THEN....)
Critical thinking	(reasoning from evidence)

What Doesn't Transfer?

Cognitive development level is not speeded up
Planning ability
Variety of individual problems: Tower of Hanoi, candle problem, balance beam and shadow problems
Reasoning skills

(Degelman et al., 1986; Clements & Gullo, 1984; Pea, 1983; Pea et al., 1985; Gorman & Bourne, 1983; Gaffney, 1984; Shaw, 1984; Rose, 1983)

TABLE 2. MIDDLE SCHOOL/JUNIOR HIGH AND HIGH SCHOOL

Skills that have and have not been found to transfer from programming to non-programming environments.

Middle School/Junior High: What Transfers?

Math word problem performance
Ability to complete designs with pieces missing
Arranging blocks to match a pattern
Estimating the picture from a set of directions
Rewriting directions in short and simple form
Creativity (design of own choice)

Middle School/Junior High: What Doesn't Transfer?

Debugging
Spatial Visualization
Line and angle estimation
Ability to do algebra
Ross Test of Higher Cognitive Processes (missing premises, questioning strategies, complex concepts, tell the middle portion of how a person solved a problem)

High School: What Doesn't Transfer? (Nothing has transferred)

Developmental cognitive level is not changed

(Dvarskas, 1983; Horton & Ryba, 1986; Milojkovic, 1983; Kurshan et al., 1983; Cheshire, 1981; Mandinach, 1986; Dalbey & Linn, 1986)

TABLE 3. COLLEGE

Skills that have not been found to transfer from programming to non-programming environments.

What Transfers?

Math word problems
Algebra word problems, males only
Word problem translation: into equations, giving correct answer
Procedure comprehension: following directions; predicting output

What Doesn't Transfer?

Logical reasoning
Visual ability
Verbal ability
Arithmetic computation
Strategies for solving word problems involving quasi-binary search

(Mayer et al, 1986; Ehrlich et al., 1984; Rucinski, 1986; Taitt, 1985)

THE ILLINOIS CAREER INFORMATION SYSTEMS HORIZONS

By: Dick McKenzie, IOICC/HORIZONS, Springfield, IL

INTRODUCTION

The Career Information System (CIS) is an interactive system for career exploration and informational retrieval. It is designed to serve a wide audience of individuals making career plans, including young people engaged in initial educational and vocational planning, school graduates undertaking the transition from school to work, and adults in career transition for physical, economic, or developmental reasons. The system was first developed in Oregon and has been implemented by fifteen states and one large city: Alaska, California (EUREKA), Colorado, Georgia, Idaho, Illinois (HORIZONS), Iowa, Maryland (ESSEX), Montana, Nebraska, New York City (Metro Guide), North Dakota, Oregon, Washington and Wyoming.

CIS COMPONENTS

The Career Information System provides substantial information on two principal elements of career development - occupations and education. These major elements have three types of components in CIS: substantive information, referral information, and the structured search.

OCCUPATIONAL INFORMATION

There are two substantive occupational information files: occupational descriptions and preparation.

- Occupational Descriptions

  The Occupational Descriptions file provides narrative statements of essential information about 301 occupations representing over 90% of Illinois employment. The description includes an overview of job functions and duties, required abilities, work settings, hiring practices, current employment, and wages. Major types of employers are also identified in this file including the military. Cross-referencing to other information about the occupation, related occupations, preparatory education and training, and sources of additional information are included. Occupational descriptions emphasize local information.

- Preparation

  The Preparation (PREP) file explains ways people prepare for the occupation. The system contains a statement for each occupation, regardless of preparation method. The Preparation statements describe the several most common methods of preparing for the occupation, any licensing requirements, tips for improving employ-ability in the occupation, and cross-references to postsecondary education and training when that is an appropriate method of preparation.

The design of CIS recognizes that the system should not attempt to be exhaustive, but should encourage interested users to examine potential occupations in several ways in order to get additional information, differing points of view, and experiences that will reveal some of the more effective aspects of various work situations. The referral information on occupations is in the occupational bibliography file.

- Bibliography

  The Bibliography (BIB) file identifies the most pertinent publications for each occupation in the system and provides information for obtaining the material.

EDUCATIONAL INFORMATION

The Career Information System provides educational information in three files: one describing educational programs in all existing fields of study, another describing postsecondary educational institutions in Illinois and the third describing colleges nationally. Referral to publications about programs is incorporated in those files.

- Programs of Study and Training

  The Programs of Study and Training (PROG) file provides descriptions of 125 postsecondary education programs. Each program includes a general description of the program area and and information about: program admission, typical coursework, graduation requirements, special things to know about the field, military training when applicable, related programs in which an individual might be interested, related occupations for which the program may provide training, community colleges, four year colleges and universities in Illinois offering the program, and vocational schools in Illinois offering the program. All school programs and the degrees available are listed by region within the state.

- Illinois Schools

  The CIS School (SCH) file contains information on over 600 Illinois schools. The school information includes: address and telephone number of the school, who to contact for more information, how to arrange to visit the school, type of school, degrees that are offered, student enrollment, accreditation held by the school, type of academic calendar, tuition, entrance and financial aid application deadlines, type of community in which the school is located, population of the city in which the school is located, and general program areas offered (Program of Study and Training file names and numbers).

  The vocational and hospital schools are listed by type of program and identified by name and address. Program information on vocational schools is presented in the Program of Study and Training files to which the schools relate.

- National Schools

  The National School (NSCH) File contains information on over 2,000 accredited institutions in the United States and its territories. All schools in the file offer programs at the four-year bachelor's level or above. Topics covered on each school in the National School File include the type of school, enrollment, tuition, admission requirements, financial aid procedures, degrees offered, and programs of study available.

STRUCTURED SEARCH

CIS was designed to be an interactive, user-friendly information retrieval system and an aid to career exploration. Therefore, people may access any relevant information at any time during their use of the system, or they may use the system's structured search components to identify topics for exploration. CIS software provides both direct access to files and structured accessing instruments.

- QUEST and SELECT

  To facilitate identification of potentially relevant occupations, the system contains coding of all occupations for twenty-one career relevant variables. These may be used selectively to configure an occupational search. Alternatively, they may be taken together as a structured interview (QUEST Questionnaire). The QUEST questionnaire has been extensively researched and field tested by people seeking to identify personally relevant occupations. People using QUEST items express preferences on topics such as the nature of work, level of abilities, degree of physical activity, amount of education and training, and wage and salary level. This component identifies occupational fields where these interests and abilities are commonly used. It enables the person to estimate the effect of various choices by generating occupational lists for which the person may wish to seek information.

- QUEST

  QUEST is an information sorting instrument, not a test. It is a guidance tool which serves to help people learn fundamental occupational concepts and to discover the relationship between their personal choices and occupational alternatives. Rather than assessing personal qualities and prescribing appropriate occupations, it is a component structured to encourage exploration of varied occupations that have attributes of interest of the individual.

- School Sort

  School Sort is a sorting instrument for educational information. It enables users to identify certain types of schools from the National School File based on seven search topics: programs of study, degree, region, school setting, school size, freshman admission requirements, and tuition and fees.

The occupational information, programs of study and school files are published in a three volume set of reference books. The QUEST access instrument and all of the data files are available in the Micro-CIS version which runs on any MS/DOS hard disk hardware with at least 10 megabytes of storage.

The system is also available in various abbreviated configurations combining software and the hard copy reference books. These are known as Micro-QUEST and are compatible with Apple IIc and Apple IIe. There is also a needlesort version.

For further information contact:

Dick McKenzie
IOICC/HORIZONS
217 East Monroe, Suite 203
Springfield, IL 62706
(217) 785-0789

MUSIC SOFTWARE - APPLE IIE

By: Scott Meland, Deerpath Junior High School, Lake Forest, IL

The Music Department of Lake Forest School District #67 is developing a computer curriculum divided into three areas:

1. **REMEDIAL** - In this area we try to target students new to the district who have not had the benefit of general music instruction or those students who have difficulty conceptualizing music theory.

2. **INSTRUMENTAL** - The instrumental approach is aimed at keyboard techniques and theory for band or orchestra students.

3. **GIFTED** - The gifted approach allows for the advanced students to experiment with more sophisticated software in the areas of theory, composition and performance.

MUSIC SOFTWARE EVALUATION

TITLE_____

AUTHOR_____

PUBLISHER_____

HARDWARE REQUIRED_____

SOFTWARE PRICE _____

REVIEWER_____DATE_____

LEVEL_____GRADE_____

PROGRAM OBJECTIVE _____

_____

POOR 1 - EXCEPTIONAL 5

CONTENT	1 2 3 4 5	SOUND QUALITY	1 2 3 4 5
EASE OF USE	1 2 3 4 5	PRESENTATION	1 2 3 4 5
DOCUMENTATION	1 2 3 4 5	FEED BACK	1 2 3 4 5

COMMENTS_____

_____

_____

_____

_____

_____

_____

RECOMMEND FOR PURCHASE YES_____ NO_____

PURCHASED _____ DATE_____

STUDENT MUSIC SOFTWARE EVALUATION

TITLE_____

AUTHOR_____

PUBLISHER_____

HARDWARE REQUIRED_____

STUDENT REVIEWER _____DATE_____

STRONGLY DISAGREE 1 - STRONGLY AGREE 5

1. I ENJOYED USING THIS PROGRAM.	1 2 3 4 5
2. I WOULD LIKE TO DO THIS AGAIN.	1 2 3 4 5
3. MY FRIENDS WOULD ENJOY THIS PROGRAM.	1 2 3 4 5
4. I LEARNED FROM THIS PROGRAM.	1 2 3 4 5

COMMENTS _____

_____

_____

_____

_____

_____

_____

_____

_____

MUSIC DEPT. 1/86

MUSIC SOFTWARE EVALUATION

TITLE SONGWRITER
AUTHOR Learningways Art Baridge and Samuel Wantman
PUBLISHER Scarborough Systems
HARDWARE REQUIRED
GRADE LEVEL 3 - 8
PROGRAM OBJECTIVE Compose a song with a piano roll technique.

	POOR 1 - EXCEPTIONAL 5									
CONTENT	1	2	3	4	5	EASE OF USE	1	2	3	4	5
DOCUMENTATION	1	2	3	4	5	PRESENTATION	1	2	3	4	5
FEED BACK	1	2	3	4	5	REC. FOR PURCHASE	1	2	3	4	5

COMMENTS: Songwriter permits user to select notes one at a time from a 2 1/2 octave keyboard and assign both duration and pitch to each note. Piano roll technique is used rather than staff and music notation. We found this unacceptable since students are not required to fit songs into any bar structure. It is difficult to transfer a melody created on Songwriter into standard notation.

MUSIC SOFTWARE EVALUATION

TITLE SPIDER EATER
AUTHOR Al Borges and Ralph Higgins
PUBLISHER KOALA WARE
HARDWARE REQUIRED Koala Pad
GRADE LEVEL 4 - 8
PROGRAM OBJECTIVE Note identification on staff and keyboard. Ear training, pitch recognition.

	POOR 1 - EXCEPTIONAL 5									
CONTENT	1	2	3	4	5	EASE OF USE	1	2	3	4	5
DOCUMENTATION	1	2	3	4	5	PRESENTATION	1	2	3	4	5
FEED BACK	1	2	3	4	5	REC. FOR PURCHASE	1	2	3	4	5

COMMENTS Presentation is not clear and instructions must be explained by instructor. Games 1 -3 are appropriate for grade school level. Games 4 - 6 require the student to recognize pitch by ear - this would develop relative pitch and interval perception.

Student reaction to the software was very good.

MUSIC SOFTWARE EVALUATION

TITLE EARLY MUSIC GAMES
AUTHOR John Paulson
PUBLISHER Counterpoint Software Inc.
HARDWARE REQUIRED _____
GRADE LEVEL 1 - 3
PROGRAM OBJECTIVE Learn note names and the piano keyboard. Create tunes and record them for play back.

	POOR 1 - EXCEPTIONAL 5		
CONTENT	1 2 3 4 5	EASE OF USE	1 2 3 4 5
DOCUMENTATION	1 2 3 4 5	PRESENTATION	1 2 3 4 5
FEED BACK	1 2 3 4 5	REC. FOR PURCHASE	1 2 3 4 5

COMMENTS: The games are enjoyable for young students and the instructional value is very elementary. Colorful presentation.

Student reaction is good.

MUSIC SOFTWARE EVALUATION

TITLE TRUMPET FINGERINGS
AUTHOR David Schwaegler
PUBLISHER Wenger
HARDWARE REQUIRED _____
GRADE LEVEL 5-8
PROGRAM OBJECTIVE Trumpet fingering drill, note name drill.

	POOR 1 - EXCEPTIONAL 5		
CONTENT	1 2 3 4 5	EASE OF USE	1 2 3 4 5
DOCUMENTATION	1 2 3 4 5	PRESENTATION	1 2 3 4 5
FEED BACK	1 2 3 4 5	REC. FOR PURCHASE	1 2 3 4 5

COMMENTS: This program is not for the rank beginner. Excellent review for the intermediate student. Allows for student record keeping and prints a copy of same.

Excellent student reaction. Very good remedial tool.

MUSIC SOFTWARE EVALUATION

TITLE KEYBOARD NOTE READING
AUTHOR Virgil Hicks
PUBLISHER Wenger
HARDWARE REQUIRED (Optional: Paddles or Joystick)
GRADE LEVEL 5 - 8
PROGRAM OBJECTIVE Identify notes on piano keyboard as they appear on the staff.

	POOR 1 - EXCEPTIONAL 5		
CONTENT	1 2 3 4 5	EASE OF USE	1 2 3 4 5
DOCUMENTATION	1 2 3 4 5	PRESENTATION	1 2 3 4 5
FEED BACK	1 2 3 4 5	REC. FOR PURCHASE	1 2 3 4 5

COMMENTS: Lessons include drills on Treble and Bass clefs, white keys, chromatic, leger lines and speed reading. Students may use the computer keyboard, paddles or a joystick. The disk contains an instructor manager which includes record keeping for 125 students. These are very good drills for piano students.

Student reaction was moderate to good. Some found the advanced levels difficult because they could not move the keys fast enough to satisfy the software.

MUSIC SOFTWARE EVALUATION

TITLE NOTE READING TUTORIAL / STAFF NOTE READING
AUTHOR Virgil Hicks
PUBLISHER Wenger
HARDWARE REQUIRED Optional (Paddles)
GRADE LEVEL 4-8
PROGRAM OBJECTIVE Introduction to the staff and note reading.

	POOR 1 - EXCEPTIONAL 5		
CONTENT	1 2 3 4 5	EASE OF USE	1 2 3 4 5
DOCUMENTATION	1 2 3 4 5	PRESENTATION	1 2 3 4 5
FEED BACK	1 2 3 4 5	REC. FOR PURCHASE	1 2 3 4 5

COMMENTS: The tutorial has a complete introduction to note reading. The staff note reading section allows the student to choose: Treble, Bass, Treble and Bass, Leger Lines, Tenor Clef, Alto Clef. The computer records the average time the student requires to answer the questions. The scores can be saved in the "Hall of Fame". The disk includes an instructor manager which can keep records for 125 students.

Student reaction was very good.

MUSIC SOFTWARE EVALUATION

TITLE MR. METRO GNOME
AUTHOR Virgil Hicks
PUBLISHER Wenger
HARDWARE REQUIRED _____
GRADE LEVEL 3 - 6
PROGRAM OBJECTIVE Learn value of notes and tap them in correct rhythm.

	POOR 1 - EXCEPTIONAL 5		
CONTENT	1 2 3 4 5	EASE OF USE	1 2 3 4 5
DOCUMENTATION	1 2 3 4 5	PRESENTATION	1 2 3 4 5
FEED BACK	1 2 3 4 5	REC. FOR PURCHASE	1 2 3 4 5

COMMENTS: This program does not demonstrate duration. Using the up and down motion of Mr. Metro Gnome's foot, the student learns where a note begins. Without the concept of a sustained sound, the program has little value.

Student reaction was poor.

MUSIC SOFTWARE EVALUATION

TITLE MUSIC BOX II
AUTHOR Virgil Hicks
PUBLISHER Wenger
HARDWARE REQUIRED _____
GRADE LEVEL 5 - 8
PROGRAM OBJECTIVE Recognition of music symbols.

	POOR 1 - EXCEPTIONAL 5		
CONTENT	1 2 3 4 5	EASE OF USE	1 2 3 4 5
DOCUMENTATION	1 2 3 4 5	PRESENTATION	1 2 3 4 5
FEED BACK	1 2 3 4 5	REC. FOR PURCHASE	1 2 3 4 5

COMMENTS: Serves as a review and helps students recall terms at a quick glance. Wrong answers are corrected at the end of each session. The game format appeals to students.

Student reaction is very good.

MUSIC SOFTWARE EVALUATION

TITLE MAESTRO SCOPE
AUTHOR Sharon Kunitz
PUBLISHER Biziwerks, Inc. / Maestro Music, Inc.
HARDWARE REQUIRED Single Disk Drive
GRADE LEVEL Grade 3 and up
PROGRAM OBJECTIVE Introductory music theory. Designed for independent study.

	POOR 1 - EXCEPTIONAL 5		
CONTENT	1 2 3 4 5	EASE OF USE	1 2 3 4 5
DOCUMENTATION	1 2 3 4 5	PRESENTATION	1 2 3 4 5
FEED BACK	1 2 3 4 5	REC. FOR PURCHASE	1 2 3 4 5

COMMENTS: Three (3) disks / 18 lessons. Sound is not optional. No printed student material is supplied, however, a course outline and score sheets are provided. The lessons are organized into three sections: four regular lessons, a fifth optional keyboard lesson and a review test. The keyboard segment is optional so that beginners with band and orchestra instruments can also use the program. It is very easy to use with excellent presentation. It would be possible to use the test sections to determine if a non-beginner could skip certain lessons.

Excellent student reaction.

MUSIC SOFTWARE EVALUATION

TITLE ELEMENTS OF MUSIC
AUTHOR Eddins and Weiss
PUBLISHER Electronic Courseware Systems
HARDWARE REQUIRED One Disk Drive
GRADE LEVEL Grade 5 - Adult
PROGRAM OBJECTIVE Drill and Test Pitch Name Identification, Key Signature Identification, Keyboard Recognition.

	POOR 1 - EXCEPTIONAL 5		
CONTENT	1 2 3 4 5	EASE OF USE	1 2 3 4 5
DOCUMENTATION	1 2 3 4 5	PRESENTATION	1 2 3 4 5
FEED BACK	1 2 3 4 5	REC. FOR PURCHASE	1 2 3 4 5

COMMENTS: A drill section and a test section are available for each of the three (3) lessons. Only students enrolled in the program and have a password may access the disk. The drills may be timed or untimed; however, the test items are all timed. The student may escape at any point and his scores are kept on the disk. This is not for a self-instructed beginner, but would be an excellent drill and test supplement.

Student reaction was very good to excellent.

MUSIC SOFTWARE EVALUATION

TITLE PRACTICAL MUSIC THEORY
AUTHOR Sandy Feldstein
PUBLISHER Alfred
HARDWARE REQUIRED Single Disk Drive
GRADE LEVEL 5 - Adult
PROGRAM OBJECTIVE Independent Theory Study

	POOR 1 - EXCEPTIONAL 5									
CONTENT	1	2	3	4	5	EASE OF USE	1	2	3	4	5
DOCUMENTATION	1	2	3	4	5	PRESENTATION	1	2	3	4	5
FEED BACK	1	2	3	4	5	REC. FOR PURCHASE	1	2	3	4	5

COMMENTS: A three (3) volume series with optional sound. Volume I - Staff, clefs, note names, time signatures, note values, rests. Volume II - Octaves, major scales, intervals, more complicated rhythms. Volume III - Minor scales, maj/min chords, inversions, progressions, transposition, writing and harmonizing melodies. Designed to be used with workbook. Each unit is short enough to complete at one sitting and it is possible to escape at any time. Each unit contains a review quiz as the final lesson. The software is well designed without major bugs.

Excellent student reaction.

MUSIC SOFTWARE EVALUATION

TITLE MUSIC THEORY
AUTHOR Linda Borry
PUBLISHER Minnesota Educational Computing Consortium (MECC0
HARDWARE REQUIRED
GRADE LEVEL 7 - 8
PROGRAM OBJECTIVE Music Theory drill on 18 different concepts. Counting, enharmonics, key signatures, terms, triads, scales, etc.

	POOR 1 - EXCEPTIONAL 5									
CONTENT	1	2	3	4	5	EASE OF USE	1	2	3	4	5
DOCUMENTATION	1	2	3	4	5	PRESENTATION	1	2	3	4	5
FEED BACK	1	2	3	4	5	REC. FOR PURCHASE	1	2	3	4	5

COMMENTS: The available programs presented in the menu are not sequentially listed, however, the manual suggests a sequence of instruction. This program is designed exclusively as a drill and all explanations would be given by the instructor. Each program allows students to choose the difficulty of the problems to be presented. The students may select exercises of increasing difficulty as skills improve. The accompanying manual is excellent.

Student reaction is very good.

SOFTWARE PRICES

Early Music Skills - $29.95
Practical Music Theory (Alfred) - $199.00
Mr. Metro Gnome (Wenger) - $49.00
Early Music Games - $29.95
Trumpet Fingerings (Wenger) - $29.00
Songwriter - $20.00
Beginning Music (Liver Burdett) - $130.00
Elements of Music (ECS) - $99.95
Maestroscope (Maestro Music) - $150.00
Music Box II (Wenger) - $59.00
Note Reading Tutorial (Wenger) - $69.00
Keyboard Note Reading (Wenger)-$69.00
Music Theory (MECC) - $49.00
Spider Eater - $10.00
Complete with Overlay

MUSIC SOFTWARE COMPANIES

Wenger Corporation
555 Park Drive
Owatonna, MN 55060
1-507-451-3010

Excellent software catalog: CODA
Listing of many software products from the major producers.

Silver Burdett
5203 Madison
Cincinnati, OH 45227

or

Silver Burdett
191 Waukegan Road
Northfield, IL 60093

Temporal Acuity Products, Inc.
Building 1, Suite 200
300 120th Ave. N.E.
Bellevue, Washington 98005

Electronic Arts
2755 Campus Drive
San Mateo, CA 94403

The Scarborough System
55 S. Broadway
Tarrytown, New York 10591

Great Wave Software
P.O. Box 5847
Stanford, CA 94305

Passport Designs, Inc.
625 Miramontes Street
Suite 103
Half Moon Bay, CA 94019

- Great software for MIDI

Decillionix
P.O. Box 70985
Sunnyvale, CA 94086

Mimetics Corp.
P.O. Box 60238 Sta. A
Palo Alto, CA 94306
(408) 741-0117

Dr. T's Music Software
66 Louise Road
Chestnut Hill, MA 02167

Merry Bee Software
815 Crest Drive
Papillion, NE 68046

Koala Technologies
3100 Patrick Henry Drive
Santa Clara, CA 95054

MAGAZINES

KEYBOARD
P.O. Box 2110
Cupertino, CA 95015-9975

ELECTRONIC MUSICIAN
5615 W. Cermak
Cicero, IL 60650

SYNTHESIZERS, MIDI, RELATED PRODUCTS

Future Sounds (Clear Covers) - Excellent products - clear plastic covers for electronic instruments.
932 East 5th Street
Oxnard, CA 93030

Korg
Westbury, NY 11590 - Electronic Instruments

Yamaha Corp.
6600 Orangethorpe Ave. - Electronic Instruments
Buena Park, CA 90620

Roland Corp. - Electronic Instruments
7200 Dominion Circle
Los Angeles, CA 90040

Ensoniq Corp. - Electronic Instruments
1 Great Valley Parkway
Malvern, PA 19355

Sequential Circuits, Inc. - Electronic Instruments
3051 N. 1st
San Jose, CA 95134

Invisible Products - Strands for Electronic Instruments
P.O. Box 230
Cambridge, MA 02139

Casio, Inc. - Electronic Instruments
175-10 Route 46 West
Fairfield, NJ 07006
(201) 575 - 7400

Maestro Music Inc.
2403 San Mateo N.E.
Suite P-12
Albuquerque, NM 87110 (505) 881-9181

Minnesota Educational Computing Consortium (MECC)
3490 N. Lexington
St. Paul, MN 55113

Sublogic
713 Edgebrook Drive
Champaign, IL 61820
(217) 359-8482

ALPHA MICROCOMPUTER VIDEODISC PROJECT FOR INTERACTIVE READING, WRITING AND COMMUNICATION

By: Keith E. Nelson, Philip M. Prinz, Deborah Dalke and Ann R. Landy

The Alpha project is the development and assessment of a method for teaching word identification and sentence production. It is designed for instruction of hearing impaired children, but is also being used with multiply handicapped children as old as twelve years and with normal, hearing preschoolers, ages two and three.

PROGRAM DESIGN

The Alpha program capitalizes on the inherent attractiveness and complex capabilities of a microcomputer system by using animated graphics as exemplars of a vocabulary of approximately 200 words. It incorporates the interactive videodisc to allow access to both still pictures and action sequences as additional exemplars for each word. A primary component of the system is a special keyboard that allows the child to activate graphics and pictures by pressing the words printed on the surface of the board.

An additional important feature of the program is signed representations for vocabulary items. By pressing a color-coded "bar" on the keyboard, the students can see the American Sign Language symbol provided from the videodisc. This insures that the representation from the child's primary language is always available.

The program is designed to supplement regular reading and writing activities. It consists of 136 lessons, each with a vocabulary of two to eleven nouns and two to seven verbs. Sentences are composed by choosing subjects, predicates and objects from the word lists. The more advanced lessons contain articles, adjectives, adverbs and conjunctions, as well. Vocabulary words are repeated across lessons to provide added reinforcement. The large number of legitimate word combinations for sentence formation provides a high degree of complexity and flexibility in comparison to other software for language instruction.

A student explores the vocabulary for each lesson as her interest directs by pressing words on the keyboard and watching the animated graphics and pictures from the videodisc that appear on the screen. By choosing words and seeing exemplars, she begins to associate the label to the concept and is ultimately able to identify the word when the exemplar is presented as a test item.

Word identification, although important, is not the core of the program. Sentence composition is the primary activity. The Alpha program can be considered a rudimentary word processor for young children. By choosing first a noun, then a verb, and finally a noun, students can compose sentences such as "The lion chases the elephant." or "The car bumps the bicycle." The computer then provides a representation of the entire sentence as a sequence of animated graphics and video pictures. The student is able to see an immediate representation of what he has written.

After sentence composition has been explored in this way, students are tested on interpretation and sentence production. An animated graphic sequence is presented on the screen to the child who identifies the words for the subject, predicate and object represented in the sequence. Sentence exploration and sentence testing allow the child to go beyond vocabulary development to more complex language skills, such as understanding of syntax.

PROGRAM FEATURES AND LANGUAGE DEVELOPMENT

The Alpha program was designed to incorporate research findings about language development in the instruction. Because whole words may be activated by pressing keys on the board, the sentence production process is much faster than it would be if the words were typewritten. By providing rapid sentence composition, the Alpha program allows the child's complete thoughts to be represented. In addition, it provides the child with repeated examples of the components of a complete thought, namely a subject, predicate and object. For hearing impaired children, in particular, such concepts are difficult to develop.

The vocabulary for the program was selected on the basis of a study of words emitted spontaneously in conversation by children between the ages of four and eight years (Hall, Linn & Nagy, 1980). The frequency with which each word was used by children was an indicator of the importance of that concept to them. Words which were moderately or very frequent (e.g. ice cream) were considered to be part of the conceptual repertoire for most children of that age span, and thus are likely to be concepts that are also in place for hearing impaired children. It was assumed that children would rapidly learn the labels for concepts that were important to them. As a result, the vocabulary in the Alpha lessons differs from those found in basal texts. Some of the words are not found in basals because they are rather long (helicopter, alligator), or are low frequency words (giraffe, kangaroo). However, all the Alpha vocabulary words were used by children whose conversations were monitored in the research.

Alpha lessons cannot be carried out independently. It is necessary for an adult to be present with the student. This characteristic of the program allows vocabulary to be incorporated in dialogue between the student and teacher. Extensive research by Dr. Keith Nelson has verified the effects of language expansion that result when children and adults converse about a topic. Specifically, when adults add information to utterances or signed messages from children and expand on the sentences provided by the child, more complex syntax occurs spontaneously in the child's dialogue. Serious language deficits for hearing impaired children have been extensively documented. Thus, this feature may be the most valuable for long term growth in language skills, such as complex sentence production and vocabulary growth through the addition of synonyms, adjectives and adverbs derived from dialogue.

PROGRAM ASSESSMENT

In addition to the instructional features, the Alpha program includes a system for automatically recording and printing all responses made by the child. This information allows the teacher to individualize instruction by selecting a lesson based on accuracy of word identification in a sentence test. Typically, the teacher first administers the test and if the record indicates that the student is identifying lesson vocabulary words at 30% to 70% accuracy, the teacher begins to use the lesson activities to improve word identification and sentence interpretation and production. On the other hand, if the printer record shows that word accuracy is 80% or better, the teacher looks for a lesson that is more challenging.

The printer provides an accurate record of progress through the lessons that are available to the teachers for their own instructional planning. Such records also allow a general assessment instructional impact.

As the description of the program features indicates, the selection of the vocabulary was based on the language production of children who are at an age when they begin to learn to read. The assumption is that word identification skills will develop rapidly with such a vocabulary. The assumptions about the effects of word and sentence exploration activities on improvement in sentence interpretation and production are based on laboratory research

and have not been verified in the classroom. Printout records generated as Alpha lessons when completed were used to test some assumptions related to instructional effectiveness.

A sample of 63 deaf children between the ages of 3 years, 4 months and 11 years, 8 months were subjects for this assessment. All children had an average or better intelligence, as indicated by performance on a scale of an accepted intelligence test. Baseline reading and language measures were administered in the preferred language mode of the child. A formal reading achievement test was also administered. Additional detail about the baseline measures may be found in other documents (c.f. Prinz, Pemberton and Nelson, 1985).

The students were divided into high, medium and low exposure groups, based on the number of lessons per week. The high group averaged 6 lessons per week, the medium group, 3 lessons and the low group, .5 lessons. The group was also divided by age into young and older groups. The age range of the young group was 3 years, 4 months to 6 years, 4 months. The range for the older group was 6 years, 6 months to 11 years, 8 months..

Before each Alpha lesson was begun, a word test was administered as a pretest measure of word identification for the vocabulary in the lesson. The computer program randomly selected animated graphics which were displayed on the screen. The student selected a word for the graphic by pressing a key on the board. Following this test, word exploration and sentence exploration exercises were carried out in which children chose words by pressing keys on the board and saw the accompanying graphics and videodisc pictures. Following these exercises, a sentence test was administered. The score was the number of words correctly identified (pressed on the keyboard) after the student had seen an animated graphic sequence on the screen. The student was asked to find the word for the subject, the predicate and the object of the sentence, although those specific terms were not used.

The records of performance accuracy on the sentence test were compared, first for young and older students, then for high, medium and low exposure groups. The initial question concerned the effect of age on sentence test performance. As expected, older students had higher average pretest scores for word identification (55% mean accuracy) than did the younger group (42% mean accuracy). This difference was significant at the .05 level. At posttest, however, there were no performance differences between older and younger students. The mean accuracy for older students was 89% and for the younger students it was 85%. The word and sentence exploration exercises compensated for an initial deficit in the younger students.

With respect to the effects of instruction time, the dependent measure in this case is a composite score of a generalized language test battery, plus the score from the Test of Early Reading or the Stanford Achievement Test. There were no significant differences in baseline scores among the high, medium and low exposure groups (Low X=39.1, Medium X=32.6, High X=35.0)., However, an effect was obtained for number of sessions per week on posttest scores. (Low X=47.6, Medium X=48.5, High X=55.1). By comparing the pretest and posttest scores it can be seen that growth occurred for all groups over the course of the instruction period, and as program exposure increased, growth also increased. This effect was significant ($F_{(2,60)} = 3.99$; $p < .025$).

A final comparison was or word identification accuracy rate for posttest sentences that were fanciful and realistic. Some of the test sentences produced by the random combination of nouns and verbs were fanciful (e.g. "The flower drops ;the rabbit"). There was some concern among teachers who participated in the project that such sentences would be

difficult for students to interpret. However, accuracy rates for the fanciful sentences were nearly identical to rates for realistic sentences. (Fanciful X=87.2; Realistic X=86.6).

The above outcomes verify that deaf children achieve mastery of word identification and sentence interpretation and production with the Alpha materials. For word posttests and sentence posttests, the children averaged between 80% and 90% accuracy. Further, their improved skills generalized to other measures of language ability. Finally, the accuracy rate on both fanciful and realistic sentences indicates that the children were relying on an understanding of English syntax to correctly identify the subjects, predicates and objects represented in the graphic sequence presented on the screen.

Additional performance data are being obtained from 30 schools, located in various cities in the U.S. and Canada, that are serving as test sites during the 1986-87 school year. These data will enable an assessment of program effectiveness for other populations in addition to the hearing impaired learner.

REFERENCES

Hall, William S., Linn, Robert L., Nagy, William E. (1980) Spoken words (Tech. Rep. No 177). Urbana: University of Illinois, Center for the Study of Reading, August, 1980.

Prinz, Philip M., Pemberton, Elizabeth, & Nelson, Keith E. (1985) The ALPHA interactive microcomputer system for teaching reading, writing and communication skills to hearing-impaired children. American Annals of the Deaf, 130, 444-461.

THE EVOLUTION OF THE REVOLUTION

The Story of Drivers' Education and Computer Literacy

By: Thomas Neudecker, Carnegie Mellon University, Pittsburgh, PA

LITERACY

Literacy means educated; it was first operationally defined as...*one who has skills in reading and writing*. Today the mastery of many knowledge areas and technologies other than language is expected. For example, basic mathematic skills have been required of the literate for some time. New technologies present themselves to society frequently; and often they become part of the current definition of literacy. The concept of computer literacy first appeared in the early 1980's when the technology was less than 25 years old and when for the first time, personal computers were being sold as household appliances. In 1982, personal computers were common gifts found beneath the Christmas tree. *From the beginning of time until 1980, there were only one million computers. Commodore alone manufactured that many computers in 1982.* [1] The personal computer momentum continued to grow, and schools at all levels began to offer courses in computer programming. This period coincided with a major shift in the United States from a heavy industry based economy to an information services based economy. Parents who worried about their children's future came to believe that computer technology would provide security in the changing economy.

This trend grew before several crucial questions about the importance of the computer were answered. Questions such as:

- What is computer literacy?
- Is mastery of this technology now a literacy skill required of all?
- Are there different definitions of literacy for different population groups?
  - If literacy has various levels of competency...
    - what is the appropriate literacy level for the professions?
    - what is the appropriate literacy level for Education?
- What technology prerequisites promote the infusion of technology into?

This paper will attempt to explore these questions through a comparison of the development of automobile technology and computer technology.

MOTORING IN 1910

At the turn of the century if one had sufficient discretionary funds they could purchase and operate an automobile. Operator's licenses were not required, but motorists did need to have skills in navigation, knowledge of the terrain, and the ability to operate and repair the vehicle. Punctured innertubes, clogged fuel and air filters, and fouled spark plugs were minor field repairs that operators expected to encounter and resolve on an outing. The often asked question *What do you do with a car*? was answered by the response *What do you know how to do*? Most motorists were confined by their lack of skills and knowledge to Sunday afternoon outings into the nearby countryside. A trip across the state by auto in 1910 was a major expedition worthy of newspaper coverage. In the early years of the automobile industry, the *power users* were those with perseverance and extramural automobile maintenance skills.

Four major developments occurred that changed the auto industry. First, the industry converged upon a generic approach to the operation of the machine. The location of the clutch, brake and throttle was standardized. The steering lever was dropped in favor of a wheel. Ignition systems, fuel octanes, and instrumentation began to converge upon a small

number of standards. This common *user interface* allows the owner of a Ford to operate a Chevrolet with a minimal amount of familiarization. [2]

The second major advancement was the reduction of the cost of the technology. Automobiles manufactured using the new assembly line technology were much less expensive than hand crafted cars. Parts were interchangeable and repair procedures were formalized. This helped to extend the affordability of the car into the middle class. By the mid 1950's, the family car became a basic part of the society. Today a once flourishing business that trained new drivers has evolved into a *rite of passage* where parents teach children who teach peers and who more than anything else, learn the rules of the road by observation. Driving skills, once extramural, became intramural skills. Today, the possession of a driver's license and the ability to operate an automobile of any manufacturer in a safe fashion is a basic literacy skill. While high schools continue to offer extracurricular driver's education courses, few students enroll. Automobile literacy is becoming universal.

At some unknown point in time, a critical mass in the use and ownership of automobiles was reached. This lead to the third major event in the auto revolution; the development of a support system, a motoring *infra structure*. [3] Governments responded to the growing numbers of car owners by paving roadways and establishing basic driving laws that were similar between the states; furthering the ease of operating a car. Business sought to capitalize upon this new market. Soon, blacksmiths began to sell fuel and perform auto repairs. Drive-in versions of restaurants, movies and banks became icons of modern society. Automobile manufacturers provided new models yearly, each bigger and better than the previous. People began to realize that those persons who used the auto for transportation rather than a horse were more productive. This *infra structure* of roads [4], auto services, and manufacturing was functional in the 1940's and was refined with the construction of the Federal interstate system during the period of 1950-1985. During this time, the numbers of cars in the United States grew dramatically to a point where two car families became common.

The use of the family car became a routine activity and one which reordered the fashion of peoples' daily life. Parents began to ferry their children to schools and other events. Shopping by catalog was replaced by a central business district serving a geographically dispersed population. The suburbs followed and flourished; the suburban shopping *Mall* then began to replace the central business district because transportation to the suburbs was possible with the family car. In short, because the auto was available, easy to use, and generally better than a horse, it became a tool.

The fourth major piece of this technology revolution was the increased power and reliability of the automobile. Cars grew from roadsters to sedans, then to station wagons, pickup trucks and onto cargo hauling tractors and semi-trailers. As the power and dependability of the technology grew, so did the support systems. No longer did drivers need to know how to repair a punctured innertube, or how to fix clogged filters, or fouled spark plugs. Adequate repair and support services existed, the roadway had fewer hazards, above all else, the machine did not fail as often. Once, a trip to the surrounding countryside required major logistics; today a trip across state is routine travel. The tool has matured into ubiquity.

The new motoring technology was changing the society. In 1910, agriculture was the principle force in the economy. Sixty-seven percent of U.S. workers were employed on farms [5], but the transition had already begun. The mechanical technology of Cyrus McCormick (reaper) and John Deere (steel plow) was coming to bear upon the farm. [6] The

development of the tractor signaled that technology, not agriculture, now commanded the economy. The Grangers and the Luddites, as dramatic as their protests were, could not stop industrialization. The family farm was being consolidated into large mechanized corporate farms. Fewer acres of crops were producing larger harvests. The age of the gasoline powered tool now was at hand. By the mid-1950's, 65% of the U.S.'s workers were employed in heavy industry. By 1979, only 3% worked on farms. [7]

COMPUTING CIRCA 1952

Computer literacy was not a topic of discussion in November of 1952, television was. This electronic technology was expanding into the middle class. For the first time Americans could watch the presidential campaign from convention to election on their home TV. On election night 1952 the general public witnessed via TV the power of commercial computing. A UNIVAC-I mainframe computer used by CBS-TV correctly predicted the election of Eisenhower over Stevenson after analyzing only 5% of the vote. [8] This event established computing as an important force in the general society.

The public was in awe of this mysterious device. It seemed as if only a small group of gifted scientists had been anointed with the specialized knowledge to understand the technology. Only these few *high priests* would be the masters of the technology, it would never be in the general public's domain. The word univac became synonymous with super intelligence, it was thought that the UNIVAC was an electronic brain. A notion quite contrary to the computer was; "**GIGO**", **G**arbage **I**n **G**arbage **O**ut.

In 1952, computer technology was less than 10 years old and the ability to use programmable instructions, *software*, rather than manually rewiring the computer to perform a new task was only one year old. Computers were very costly and belonged to a very limited domain of researchers. Thomas Watson Sr., then President of the successful International Business Machine Corporation, believed that fewer than ten computers would be required in the entire world. When he rejected a proposal to work with the developers of UNIVAC to develop a commercial computer company, he said in a memo; *There can be no reasonable interaction possible between Eckert-Mauchly and IBM.* [9] Eckert and Mauchly then sold their technology to Remington-Rand.

At this time, computer operating systems, the set of instructions that tell the computer how to perform basic tasks, varied from one machine to another; even between those computers made by the same manufacturer. Computers did not share a common user interface, as the automobile does. Software, those written codes that instruct the machine how to perform complex tasks, would only work on identical computers. Imagine attempting to memorize, from a poorly written manual, the correct speed limit for all of the streets upon which you drive. In this environment, automobile traffic tickets would be as common as computer *Syntax Errors*.

Only a few computer installations were in existence then, and most were hidden by secrecy clauses required by the military, who funded these projects. Predictably, almost no interchange existed between sites and very little *infra structure* developed.

Even with all of these limitations the power of these computers were remarkable. They were able to perform almost 1000 floating point mathematical instructions per second (0.001 MIPS) [10] at a cost of $400,000. [11] The 1952 computer technology had power, [12] but few could utilize it because it was very expensive and lacked a common, intuitive *user interface*. Because there were few users due to the high level of extramural skills required to operate a computer, there was not *infra structure* of support.

COMPUTING IN 1975...THE REVOLUTION ACCELERATES

To participate in computing before 1975 required the sponsorship of a major research university or of a corporation with very deep pockets and a keen eye to the future. In 1975, the Altair microcomputer and a new version of John Kemeney's mainframe programming language BASIC (rewritten by Bill Gates then a college freshman and now the president and CEO of MICRO-SOFT) was introduced. The era of the *computer hobbyists* had begun.

The Altair, the first microcomputer, begat the Mark 8, which promoted the IMSAIL, then came the Apple II and Commodore PET and the Radio Shack TRS-80 into the marketplace. Many, many, more personal computers followed. These home computers were created by start-up companies of which, some have survived and of which most have not.[13] If you, in 1975, had sufficient disposable income, somewhere between $800 to $2000, depending upon which *peripherals* (components) you selected, you could purchase a personal computer. During this beginning phase the entire world of microcomputers was initially devoted to the hobbyists, primarily HAM radio operators. These hobbyists survived by virtue of their skills in electronics, programming, and eagerness to embrace and learn the rapidly changing technology (with all of its complexities).

WHAT DO YOU DO WITH IT?

What can you do with it, the question asked of the 1910 motorist was the same question asked of the 1975 computerist. The answer to this question was the same in both time periods...*it depends upon the user's extramural skills*. The 1975 computer user needed to learn the operating system's command language, elementary programming skills, and basic hardware trouble-shooting and repair. Just as the early motorists had to learn to clean fouled spark plugs, the computerist needed to know how to set IC-Chips into their sockets.[14] Those who did master these skills were successful in maintaining systems and often in developing commercial and public domain software. These pioneering personal computer programmers discovered that the relatively small 8-bit processors were slow and RAM was expensive. The programmers used tricks and other unusual coding practices to obtain better performance. Nuances in the architecture of CPU were exploited, at a cost. Because of CPU specific coding, the programs written were not portable from one machine to another. [15] Programmers would have to recode a program to work on another computer and new tricks would have to be found.

PERSONAL COMPUTER PROGRESS CONTINUES

In October of 1979, a business forecasting program called VISICALC was introduced by two Harvard students. [16] The program was a electronic version of the generic pen and paper spread sheet. In its computerized form, VISICALC allowed for the rapid recalculation of several "what if scenarios". VISICALC, written for the Apple II computer (and converted later for the other principle micros) helped to establish Apple and it did more than anything else to "legitimize both the microcomputer and electronic spreadsheets" [17] in the world of business. Almost simultaneously a simple word processing program, called *EasyWriter* , written by the notorious telephone phreak John, *Captain Crunch*, Draper hit the market adding to the productivity of the Apple II.[18]

The combination of the relatively low cost microcomputer and useful software created a new group of computer users far removed from the *high priests* of the 1950's. Unlike their predecessors who sought to master the machine, these users were satisfied to become *application literate*. They were skilled in the use of a limited number of productivity programs such as spreadsheets, data bases, and word processors. This software fueled the growth of the personal computers industry and education. The dropping price of hardware and its increasing performance further encouraged the growth of the computer industry.

GROWTH OF THE COMPUTER INFRA-STRUCTURE

In 1979, one year after the appearance of the Apple II and three years prior to the introduction of the IBM-PC, computer scientist Allen Newell noted that:

> *...Because a personal computer is dedicated to a single user, service is not degraded by other simultaneous usage...personal computers in the late 1980's will be capable of performing one million mathematic instructions per second, have one mega-byte of RAM, and be capable of displaying graphics in a resolution of one million pixels* (dubbed in the trade press as a 3-M computer).[19]

A new *infra structure* would be needed to support and to exploit the power of these advanced function workstations. In 1982, IBM and Carnegie Mellon University initiated a joint research and development project to design and implement a new user-interface and file distribution system for a large scale (10,000+ nodes) network of advanced function workstations.[20] Project Andrew sought to design a intuitive, common user interface that would be portable from one vendor's computers to another. The Andrew interface followed many of the design standard developed at Xerox's Palo Alto Research Center (PARC). *No Xerox product was itself a sufficient commercial success for PARC. None had a direct impact on the sales of personal computers. That was left to Steven Jobs, then with Apple Computers...[who] brought out the LISA in 1983 and the much lower-priced Macintosh in 1984, using the new technology in many of the same ways as did PARC.* [21] A common user interface is only one of Andrew's features. The Andrew standard [22] allows programs written on one vendor's computer to run on anothers. Just as rail gauge standards made boxcars portable from one railroad to another, Andrew makes software portable from one type of computer to another [23]. Today [24] Andrew is running on IBM RT's, DEC Micro VAX computers, and SUN Workstations.

As predicted by Newell in 1979, the computer industry improved the power of its computers from 8-bit processors to 16-bit, then to 32-bit, and on to RISC [25] architectures. Several manufacturers offered machines of the 3-M+ class by 1986. By 1988, these engines are expected to be available at an educational or volume price of $3000.[26,27] Others now predict that the power of the super computer, CRAY-1, will be available in an affordable desktop version by 1990.

WHY COMPUTE?

Tools are used by all of society's occupations and professions. Farmers, machinists, and information processors routinely use a repertoire of tools. Many professions use identical tools for differing tasks. Each profession strives to adapt the day's newest and best technology to their own needs. Most farmers accepted the tractor as a replacement for the draft horse. Those who did not, failed. Auto workers have learned to accept robots and now work servicing and repairing these computer controlled welders and painters. Job tasks that have not modernized, have been lost in the marketplace to cheaper labor or to those who have increased production with new technology. Computer automated manufacturing transformed the U.S.'s economy. Times of economic transition are times of opportunity. Just as the Fords, Firestones, and Carnegies of the industrial revolution reordered the wealth of the nation through creativity, the entrepreneur of the high-tech age, the Watsons, Shockey, Jobs, and others to follow will realize great opportunity. [28]

Data processing accounted for 46% of the U.S. gross national product in 1979.[29] Those who fail to adopt this new technology will not capitalize upon the new opportunities. The

U.S. may soon, indeed, be an "average" country if it misses the opportunity to multiply the productive and creative force of its' people with the new tools of computer science.[30]

THE LITERACY SOLUTION

In 1982, computer literacy was a concept without a generally accepted definition. Curricula in schools at all levels seemed to assume that computer literacy is equated to programming skills. Courses in programming languages, such as LOGO in the elementary grades, and BASIC and PASCAL in the secondary grades, were common in the early 1980's. Higher education responded in a similar fashion. Computer science programs and information sciences programs appeared and flourished. For the most part, these curriculums also focused upon programming skills. Often they were staffed with faculty with limited educational background and experience with the new technology.

These programming curricula were introduced into schools because of cost factors. The schools, generally, owned very few personal computers or had very few time-sharing job slots on centralized computing resources. Just as a critical number of cars needed to exist to fuel the development of support systems, so too do a significant number of computers need to be available to spark the development of support systems, such as electronic mail and data bases and highly specialized software.

Programming is an instructional-end unto itself. It is appropriate to teach programming to select groups of students who demonstrate the aptitude and have the prerequisite math skills. Students in programming classes can be easily scheduled into computer labs to complete assignments. The computing resources needed can be predicted and managed. On the other hand, if the computer is considered to be a general problem solving tool that utilizes such software capabilities as word processing, data bases, graphics, and spreadsheets, it must be available to the student or professional whenever they encounter a problem whose solution would be rendered with the technology. The tool must be available when needed. Schools must provide near immediate access to the resource if this level of literacy is to be achieved and supported. Additionally, all of the faculty must support the student's use of the technology by being able and being available to assist students and colleagues. If not the tool will not be adopted. A lab of 20 computers can serve many students learning only to program, but it can serve very few students who use the equipment as a general tool for generic professional tasks such as word processing, data retrieval, budgeting and the like.

LEVELS OF LITERACY

Today the operational definition of literacy in higher education is evolving beyond the domains of computer science and mathematics. Programming skills are not the only component of computer literacy. Today, powerful microcomputer applications such as charting, statistical analysis, word processing, desk top publishing, spreadsheets, spelling checkers, telecommunications, and relational databases exist. Literacy is now defined in terms of these software applications.

These programs replace the typewriter, note card filing systems, budget sheets, and complement mail and telephone systems. These computer programs assist the professional to communicate, to share his knowledge and information. Charts and other presentation graphics reveal new insights to data. Statistical analysis test the significance of assumptions, and telecommunications allow the new information to be electronically distributed to other computers connected via telephone, satellite, or other electronic links. These are the computer tools of today's professionals. Computer literacy for today's professional is measured by the user's competency in the use of these tools.

A factor limiting the growth of literacy is the difficulty in learning to use several different computer operating systems. Research in the language arts indicates that the average person has the capability to obtain literacy in only 1.5 conversational languages or by extrapolation 1.5 operating systems and/or programming languages. A tool should assist the practitioner, not be an impediment to their performance. What is required of new computers is a common, intuitive user interface that will allow the users of one vendor's system to easily move to another vendor's computer. This also requires that software be transportable from one system to another, just as one's professional tools are transportable from one job site to another.

PATHS TO LITERACY

What path will lead to computer literacy? The same developments that lead the automobile industry to establish motoring literacy will also promote computer literacy. First, there must be enough computers present so that students and professionals have reasonable access to the equipment. The computers must have sufficient power to support emerging knowledge base applications. This will happen when the cost of advanced function computer workstations, the so called 3-M class workstation, reaches an affordable price range.

Secondly, an intuitive, easy to use, common operating system is required. Today's computer systems are just too difficult to master. Users must be able to sit down at a computer made by any number of manufacturers and be able to solve problems and produce information with only minimal familiarization, just as we can drive a car whether it was manufactured by Ford or General Motors. Software programs and their data files must be transportable from one vendor's machine to another just as we can drive to new destinations over unfamiliar roads. Otherwise, users will become machine or operating system literate not technology literate. This would be similar to requiring separate drivers' licenses for each make and model of automobile you drive. Or imagine that separate interstate highways needed to be constructed for each of the different brands of automobiles. The fact that Fords, Studebakers, Packards and others could all use the same infra structure of highways promoted the automobile technology.

Data needs to be integrated and centrally coordinated to encourage the rapid electronic exchange of information. These computer advantages are but the precursors to the real power of computers found when intelligent systems are developed.

KNOWLEDGE BASE APPLICATIONS

Artificial intelligence technology is the next major development in data processing. Major research projects are underway in the United States and Japan.[31] Technology research currently being tested suggests that several artificial intelligence applications for education are now possible. The realization of these programs is waiting only upon funding for application development, not for basic research funding. For example, existing technology can create artificial intelligence programs that emulate professional judgements in the domains of internal medicine, oil prospecting, and automated manufacturing.[32] Intelligence tutors that significantly improve student learning in the domain of geometry and in learning to program in the computer language LISP are in use today.

SUMMARY

Literacy is generally a measure of an individual's mastery of a skill or knowledge area. Discussions, today, of computer literacy in the professions are really discussions of the infusion of computer technology in the professions. Literacy in any professional skill really can not be discussed until the skill is required of all professionals. It is the position of this paper that the generic computer applications of word processing, spreadsheets,

databases, communications, and presentation graphics are basic productivity tools that enhance professional performance. The mastery of these applications will become the initial standard of computer literacy in the human services. This first standard will become universal when the technology becomes affordable, powerful, easy to use, and when a supporting infra structure is created. Professional education will play the key role in transition to computer literacy by providing base technology education, remedial training to existing professionals, and by developing future technology applications.

---

[1] Naisbitt, John, 1982. *Megatrends: Ten New Directions for Transforming Our Lives.* New York: Warner. Page 25.

[2] *Note the behavioral differences between the verb forms "familiarization" and "to learn".*

[3] *This analogy of the development of the "computing infra structure" and the development of the automobile in society was first proposed by James H. Morris in an unpublished paper,* ***Thoughts on an Information Utility*** *, (page 10), June 24, 1986, Carnegie Mellon University.*

[4] *It is important to note that with few exceptions this infra structure supports all automobiles Fords and Chevrolets can both use the same roads and fuel. During the early stages in the railroad system the track gauge varied among the companies. A boxcar from one company could not be switched to another's engine. When common rail standards were adopted portability was achieved. A boxcar could then be coupled to an engine from any company.* This provided both the shipper and the railroad with expanded markets.

[5] Neudecker, Tom and Patricia Burke; **Parents, Children and Change**, edited by L. Arnold, Lexington Books/D.C. Heath Books, Lexington, Mass. Page 20

[6] *Other technologies influencing the time were crop rotation, land fertilization, scientific stock breeding, and regional production of products.*

[7] Neudecker, Tom and Patricia Burke; **Parents, Children and Change**, edited by L. Arnold, Lexington Books/D.C. Heath Books, Lexington, Mass. Page 20.

[8] Shelly & Cashman, **Introduction to Computers and Data Processing**, Anaheim Publishing Company, 1980, Page 2.7

[9] Shelly & Cashman, **Introduction to Computers and Data Processing**, Anaheim Publishing Company, 1980, Page 2.8

[10] *MIPS is the acronym for one Million Instructions (floating point math instructions) Per Second.*

[11] Shelly & Cashman, *Introduction to Computers and Data Processing*, Anaheim Publishing Company, 1980, Page 2.3

[12] *Today's home computers are capable of preforming 500,000 floating point mathematics instructions per second (.5 MIPS) for a cost of less than $1500.*

[13] *Companies such as Cromemco, Sphere, Kentucky Fried Computers, North Star, Proc Tech, and Osborne are names from the past remembered by trivia experts.*

[14] *It was also common during the period of 1950 to 1960 for owners of television sets to remove tubes from malfunctioning sets to be checked for possible defects. Corner drug stores often provided the test equipment and sold replacement tubes.*

[15] *The CP/M operating system written by Gary Kildall in 1973 was an attempt to bring portability to software written for microcomputers based upon the Z-80 series of CPUs. CP/M had a large library of public domain applications and was moderately successful until the major market shift to Micro-Soft's Disk Operating System and the new IBM-PC in 1982.*

[16] *"VISICALC" was written by Dan Fylstra and Dan Bricklin, who formed the "Personal Software Company."*

[17] Freiberger, Paul, and Michael Swaine, **Fire in the Valley**, Osbourne/McGraw-Hill, 1984, Pages 229-230.

[18] Freiberger, Paul, and Michael Swaine, **Fire in the Valley**, Osbourne/McGraw-Hill, 1984, Page 230.

[19] Newell, Allen; *The Future of Computing* at Carnegie Mellon University, Unpublished report, Carnegie Mellon University, Pittsburgh, Pennsylvania, Page 2., 1982.

[20] *The Information Technology Center and its project* ***Andrew*** *is funded by IBM.*

[21] Crecine, John P., *The Next Generation of Personal Computers*, **Science**, February 28, 1986, Vol. 231, pp. 935-943.

[22] *IBM has announced that* ***Andrew system software*** *will be a licensed product to the education market in the second half in 1987.*

[23] *Early Rail Roads used differing width, or gauge, rails the trucks, wheel assemblies, on box cars was fixed. Box cars from one gauge system could run on another. The Rail industry after many years of competition standardized upon a common rail gauge. This allowed a box car from one rail road to run on another.*

[24] *Today, January 1987, Andrew is operational on these workstations. By publication date of this book it is expected that Andrew will be operational on other 3-M workstations manufactured by other vendors.*

[25] RISC is the acronym for **R**educed **I**nstruction **S**et **C**omputer.

[26] *In Science, February 28, 1986, Vol. 231, pp. 935-943, John P. Crecine suggests that such engines will be on the education market for $3000.*

[27] *The 3-M standard has grown to what is now a 10-M machine (increased processor speed and RAM) but the expected cost has remained the same for the educational market. See NeXT, Inc. promotional brochure page 7 ©1986.*

[28] Neudecker, Tom and Patricia Burke; **Parents, Children and Change**, edited by L. Arnold, Lexington Books/D.C. Heath Books, Lexington, Mass. Page 25.

[29] Neudecker, Tom and Patricia Burke; **Parents, Children and Change**, edited by L. Arnold, Lexington Books/D.C. Heath Books, Lexington, Mass. Page 20.

[30] Servan-Schreiber, J.J. and Crecine, Barbra, **The Knowledge Revolution**, Carnegie Mellon Press, 1986, Page 14.
36 See Feigenbaum, Edward A. and McCorduck, Pamela; ***The Fifth Generation: Artificial Intelligence and Japan's Computer Challenge to the World***: Addison-Wesley Pub. Co., 1983. 8303 ix, 275 p. ; 24

[32] *See the many references to the Cadeusus Project, an internal medicine expert system developed at the University of Pittsburgh or the DENDRAL or MYCIN Programs developed to prescribe medicine dosages.*

GOALS AND MEANS IN EDUCATION
(Notes by a Piagetian Software Author)

By: Thomas C. O'Brien, Southern Illinois UN at Edwardsville, Edwardsville, IL

(NOTE: The article, Computers In Education: A Piagetian Perspective, appeared in the March, 1987, Vol. #66, Number 4, issue of *Principal* magazine and is appearing here with the permission of both the author and the magazine.)

What's so special about computers in education? Their interactivity. You do something and the computer does something back. But wait a moment. Let's start at the beginning.

In a recent article (Note 1), I asked about the goals of education. It seemed to me that American education has commonly asked the question, "How do we set there?" without first asking the question, "Where do we want to go?" It's as though we argued whether we should take a bike, a bus or a a boat without having first worked out a destination.

Before deciding on bikes, buses or boats, let's discuss destinations. The present goals of education -- at primary, secondary, and tertiary levels -- are largely concerned with static knowledge; that is, with the reception and storage of information and associations. (Wyoming? Cheyenne. Colorado? Denver. North Dakota? Oops, I forget.) This is especially true in mathematics education. At the early elementary school level, mathematics education largely amounts to the memorization of nonsense syllables (e.g. 4 x 5 = 20) from flash cards with no attention to relationships, context, utility or generalizability. As the grade level gets higher, the situation is much the same except that the nonsense syllables involve more steps. "Dead Mice Smell Bad" is a major issue in sixth grade math. What? Yes. Divide. Multiply. Subtract. Bring down. That's "mathematics"! That's how you do long division.

Need it be mentioned that present day computers can perform two billion calculations in one second? Need it be mentioned that present day newborns will be high school graduates in the Twenty First Century and that some of their children will be alive in the Twenty Second century? Need it be mentioned that the procedures of math from arithmetic calculation through matrix manipulation to analysis of covariance and beyond, are now available through computer software at a cost of a few dollars or, in less glossy public domain format, free?

In the previous article and in a number of other places (Note 2), I have suggested that the main goal of education is the development of complex thinking rather than the storing of inert associations and routine procedures. People, children especially, are active knowers and pursuers of knowing. Watch a two year old if you have any doubts. They are not mere storehouses for someone else's facts.

But the purpose of this article is to address means, not goals, in education.

Consistent with present goals -- ie. static knowledge -- transmission, repetition, and reward are the principal means of education. The child is seen as a passive entity. Knowledge is seen as "stuff" to be put into storage and brought up at test time.

Teachers focus on more and more efficient ways to transmit associations. Children repeat them endlessly. Teachers reward children (with smiley faces and M&M's), thus wresting control from children and making them more and more passive. And, administrators promote triviality and line paper peddlers' pockets by insisting that we all focus in numerical data which measures children's knowledge of the nonsense syllables they've been taught.

The problem is that it works! Organisms adapt. Children learn the game, adapt to it, and we are all the poorer. It is the rare teacher above fourth grade -- and I meet hundreds, perhaps thousands, of teachers each year -- who reports that their kids are eager, inquisitive, willing to tackle new problems. (The origin of "Student" is the Latin "studere," to be eager.) What I hear and see is that kids are b-o-r-e-d. What I hear and see, from fourth grade through graduate school, is that students won't tackle an original problem.

Yes, it works. We can drum thinking out of kids and we can succumb to absurd headline-grabbing reports (Note 3), that we should spend more and more time doing so.

How did such a situation come about? Much of American education -- goals and means -- is derived directly from the assembly lines of the rapidly industrializing America of 1910-1920. Under fire, then as now, from newspapers, parents, school boards and societal fashions, American school administrators embraced industry's assembly-line -- in particular that of the Ford Motor Company -- as a model for education.

In a history so fully researched that the reader is tempted to cry "Stop, My head hurts.", the historian Raymond Callahan documents the domination of American education by factory-oriented thinking centered about three themes: 1) mass production policies and practices, 2) cost-effectiveness, and 3) time/motion "efficiency" research. (One study involved the carrying of pig iron from one part of a foundry to another. Schmidt, the subject of the research, increased his daily haul from 12 1/2 to 48 tons when his salary was upped from $1.15 to $1.85 per day.) (Note 4)

There's a second thread to the fiber of present-day American education: the psychological research on which much of education is based. Many of the studies involve brute memory. In much of the research, meaning is deliberately banished from investigation -- as in nonsense syllable research -- in order to insure "scientific" cleanliness. Perhaps most important, much of the research is concerned with performance under various EXTERNAL conditions of reward and punishment. And finally, a shockingly large amount of the research involves animals -- pigeons, dogs, even worms -- rather than children.

But a third thread in the fabric of American education is perhaps the weakest and most disturbing. American education, at least in my twenty-five years of involvement, has seen a parade of fads and bandwagons. Remember the Initial Teaching Alphabet? Programmed instruction? Modern math? Behavioral objectives? SMSG? Performance-based contracting? The open classroom? IPI? The talking typewriter? Each of these promised to be the be-all and end-all of educational solutions and the fact that none of them lasted attests to educators' short attention span, to their unwillingness to look at underlying issues, and to the resiliency of the factory model, the knowledge-as-coal-to-be-stored view, the stimulus-response approach to education.

What does this all have to do with computers? At the most obvious level, computers are a contemporary bandwagon in American education. Are they different from what we've seen? Just another fad?

Given the increasing role of computers in the American work space and in society in general, the answer is probably no. But this is a two edged sword. One definition of computer is "Ten Thousand Idiot Clerks working at the speed of light." (I wish I could cite the author. What a lovely phrase!) The harsh side of the sword is that computers will do foolish things - effortlessly and speedily if we ask them to do so. They can deliver nonsense syllables by the megabyte.

Earlier I said that school learning was largely a matter of transmission, repetition and reward. Certainly this squares with most of your formal educational experiences, I suspect, but I'll bet that it doesn't fit your "real life" learning. Is your knowledge of your neighborhood, or your spouse, your favorite restaurant, or your children -- name the issue - attributable to transmission, repetition and external reward?

There is, in fact, a body of research quite different from the brute memory/nonsense syllable/external reward tradition. Sixty years of research by Jean Piaget and colleagues, widely known by educators and even laymen throughout the world, shows knowledge to be an active construction, a growing fabric of ideas and relationships, not an inert lump of coal.

In this view, the fabric interacts with the outside world. It acts on the world (Think of yourself screening, interpreting and understanding the writings of R. M. Nixon or Tacitus or E. B. White) and the outside world acts on it, modifying the fabric to incorporate the new reality. Knowledge, contrary to the nonsense syllable approach, is not additive, as with bricks in a wall. Rather, ideas evolve. Your notion of vacation and Kiev and sex are intrinsically different from what they were five years ago, I'll bet.

In this view, the mind is not a camera. The human mind constantly organizes reality into fabrics and these fabrics tend toward coherence, stability, economy and generalizability.

From the educator's point of view, two other aspects are worth noting: The fabrics of a six-year-old are smaller and less numerous than those of most adults (though they are incredibly rich). This is obvious and rather trivial. Adults have lived longer and so their knowledge is quantitatively greater than that of most (but not all) children. Present day education recognizes this and sees its role as adding to the knowledge of children -- filling their cup fuller. But far more important and widely unattended to is the fact that children's fabrics are qualitatively different -- not only smaller -- from ours. That is, their cup is not only less full; it contains different stuff. I will not touch on this issue except to encourage readers to look in on Ed Labinowicz's two books (Note 5) or to read any of our 26 issues of SEEDBED. (Note 6)

Second, the fabrics (organized complexes of ideas) tend not merely to interact with reality. They quest. Have you ever watched an infant coordinating her fabrics of balance, muscle, and intention to learn to grasp or to walk? Those first motions -- primitive, tentative and full of what adults might call "errors" -- evolve into the effortless coordinations that you and I now take for granted as we walk about. Further, crazy as it may seem, people who learn to walk quest even further. They run. They play tennis. They take part in triathalon contests.

How do these fabrics grow? They are pre-wired for growth, provided the interactions permitting and provoking growth are available. If the conditions permitting extension and elaboration of fabric are available, learning takes place. If not, growth is stunted. In this view, education is mental nutrition.

From this point of view, the key to learning is dissonance. What is dissonance? A mismatch between where we are and where we want/need to be. The dissonance must involve moderate mismatch with the fabric. If it's too small, boredom sets in. (2 + 2 is a trivial problem for you; it is not for a four year old with a pile of chips in front of him.) If the dissonance is too great -- differential equations are too complex for the average 12 year old (as are fractions, research shows!) -- there's more boredom and a return to arenas in which moderate mismatch obtains.

But given moderate dissonance, one acts as though pre-wired. It's almost irresistible. When did you last slow down to look at a fender-bender in commuter traffic? It's undeniable that dissonance is engaging. That's why high-rise beacon lights flash at random intervals.

When one engages a dissonant event/idea/person/place/(etc.), interaction takes place. You act on the dissonant "thing" and it acts upon you. J. McV. Hunt likened the issue to eating steak. You act on the steak and the steak acts on you. (Note 7)

There are many aspects of dissonance, all of which lend themselves to the educational situation. Think of your own learning in a wide variety of situations as you read.

One major aspect of dissonance -- widely employed by present schooling -- is competition. But what about self-competition? What about challenge? Surprise? Regularity? Randomness? Novelty? Mystery? Uncertainty? Complexity? Ambiguity? What of the built-in dissonance-detectors, curiosity and quest?

In many fields of scholarship there are primitives -- in physics: pions and muons and mesons. The aspects of dissonance above are educational primitives. Where in the educational literature, or in teachers' guides, has attention been given to dissonance -- to uncertainty and ambiguity, for example -- except to warn against such stuff?

All this said -- I have been saying much the same thing for nearly twenty years -- what's new? What's new is the availability of microcomputers in American classrooms. There are at least five important aspects to this new educational resource and as most things in life, they come in "good news and bad news" form.

1. The bad news first. As I've said, computers can deliver educational trash -- mind-killing drill and practice on nonsense syllables -- effortlessly, patiently and, in some cases, with a patina of scientific respectability, thanks to elaborate record keeping and data-reporting schemes. (Perhaps the results should be reported as they actually are: "John Smith memorized 23 of 27 nonsense syllables for a score of 85.185%, which gives him 87.301 percentile rank with other nonsense-syllable memorizers in the class, 83.109 percentile in his state and 83.109 percentile in the nation.")

2. On the bright side, the availability of micros makes it clear how trivial the algorithm-dominated mathematics curriculum really is. Here are the first twenty five cube roots of numbers starting with 1,000. I dumped them out in seconds with a two line program. The computer printed the results faster than my eye could follow.

1000	10
1001	10.0033322
1002	10.0066622
1003	10.00999
1004	10.0133156
1005	10.016639
1006	10.0199601
1007	10.0232791
1008	10.0265959
1009	10.0299105
1010	10.0332228
1011	10.036533
1012	10.0398411
1013	10.0431469
1014	10.0464506
1015	10.0497521
1016	10.0530514
1017	10.0563486
1018	10.0596436
1019	10.0629364
1020	10.0662271
1021	10.0695156
1022	10.072802
1023	10.0760863
1024	10.0793684
1025	10.0826484

I am told that at the turn of the century, Hungarian schools required hand-calculation of cube roots for secondary school graduation. Are we doing similar things today?

3. More important, widely available computer programs enable teachers to address familiar topics more easily or more richly than could possibly be the case with fixed-text (as distinct from interactive) classroom treatment.

3.1 For example, Les Dice (a public domain program by the Belgian Willy Van Hamme (+) available in Apple format from the Teachers' Center Project) enables students to investigate the tossing of 1-6 dice. The program shows the tosses and compiles a histogram of actual and expected frequencies for each experiment. What's the most frequent sum if six dice are tossed?

3.2 Mathgrapher (previously titled Graphmaster) is a graph drawing program available from HRM, Pleasantville, NY. One of many such programs, it's my favorite because it was written by a 16 year old, Steven Cohen, a student of Dr. Irwin Hoffman, George Washington High School, Denver, CO.

Want to investigate the role of "m" in y = mx + b or "c" in y = ax^2 + bx + c or "n" in y = x^n? Take a look at graphs produced in seconds at the touch of a few keys.

Y=X^X
X-AXIS: 0...12 STEP 1
Y-AXIS: 0...12 STEP 1

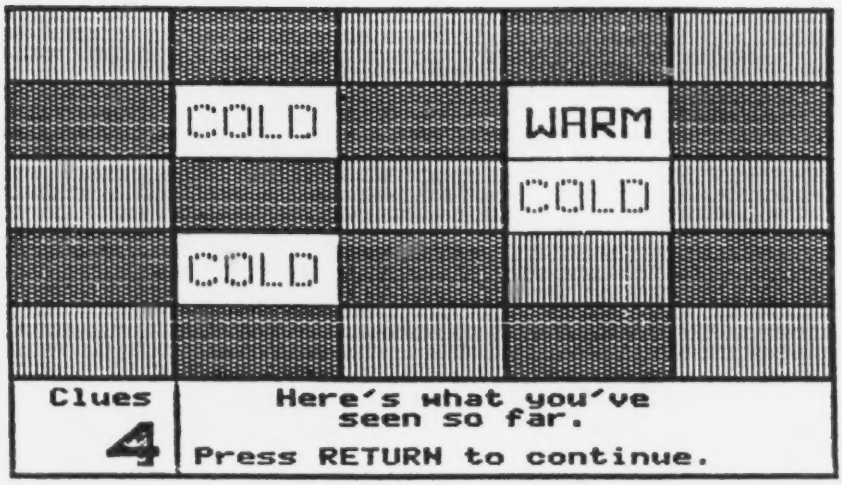
3.3 The Geometric Supposer is a fabulous playground for geometry inquiry. The four-part series, available from Sunburst Communications, Pleasantville, NY, enables children to investigate questions such as "What sort of quadrilaterals are circumscribable? Inscribable?" What happens if you draw the angle bisectors of a triangle? Does this always hold? What about quadrilaterals? Why does side-angle-side insure triangle congruence? What is the analog for quadrilaterals?

3.4 Tired of hand calculations? Try muMath (MicroSoft, Bellevue, WA), which solves equations, differentiates and integrates functions, performs matrix operations, and in general carries out mathematical algorithms with blurring speed and accuracy.

3.5 At this writing, there are several hundred pieces of exemplary education software, much of it unknown to teachers. Several lists of such software are given below. (Note 8) Clearly the computer-in-education movement is no longer in its baby shoes.

4. But a more important aspect of computers-in-education is that they bring new issues to the classroom, issues which math teachers have often wanted to address but couldn't because of lack of external support -- print publishers' conservatism, the domination of standardized tests, and the lack of systematic channels for the dissemination of new ideas.

For example, inference is a two-fold and major aspect of one's fabric of ideas. It both provides "glue" to hold the fabric together and it enables the fabric to engage new ideas and grow. Thanks to computer software, inference can be a major component of the curriculum as early as grade 1. Safari Search (Sunburst Communications, Pleasantville, NY) poses twelve inference activities to challenge learners from age 5 through adult. A Flamingo is hiding in one of the 25 boxes. If the box you choose touches the Flamingo's box sidewise, you are hot. If cornerwise, warm. If the box you choose does not touch the Flamingo's box, you are cold. Can you find the Flamingo?



Baffles (Conduit, Oakdale IA) has been used richly with fifth and sixth graders -- even with "low-ability" children -- to the extent that they refuse to go to recess when engaged in investigation. Again concerned with necessary inference, Baffles was designed for university pre-med students to develop hypothesis generating, hypothesis testing, and inference abilities.

Although plausible inference is a major aspect of the glue which holds one's mental fabric together and extends it, plausible inference ("There's good, but not sufficient evidence that such-and-such is true.") gets virtually no attention in the present school curriculum. The King's Rule and Royal Rules (Sunburst Communications, Pleasantville, NY) brings these issues to children in a hypothesis-testing activity in which students gather evidence, generate and test hypotheses, and infer with mounting, but not necessary certainty. The activities are based on the research of P. C. Wason, University College, University of London, on the construction and elimination of hypotheses.

5. Finally, more generally, and more important than all the details above, is the notion that computers can revolutionize our approach to the central aspect of education: children's learning. In place of the mechanistic, computational, behavioristic approach so widespread -- almost wholly concerned with the transmission of static knowledge and with repetition, recall and reward -- computers can enable children to be active in the construction of knowledge. Existing software, widely available, embodies richly the heart of the learning situation, dissonance. In terms of challenge, surprise, mystery, discernment of regularity and irregularity, novelty, uncertainty and complexity, they are to textbooks as flourishing rosebushes are to hardwood floors.

Earlier I said that dissonance is the key to learning. Organisms adapt through interaction with situations involving moderate mismatch with their present state. If American education can be regarded as an organism -- and I dearly trust that it can (else it is lifeless), the moderate mismatch provided now by the computers-in-education movement will enable traditional education to construct a fabric based on what's important and what's nutritious in education, quite a different view from that obtained sixty years ago from the assembly lines of the Ford Motor Company.

NOTE 1:

O'Brien, Thomas C. "What's Basic? A Constructivist View," Basic Skills, Issues and Choices, first of two books, April, 1982.

NOTE 2:

O'Brien, Thomas C. "Why Teach Mathematics," The Elementary School Journal, February 1973.

O'Brien, Thomas C. "New Goals for Mathematics Education," Childhood Education, February 1974.

O'Brien, Thomas C. "Implications of Piagetian Research for Teachers of Teachers and Teachers of Children."

O'Brien, Thomas C. "Some Observations for Education," The Genetic Epistemologist, Vol. X, No. 5, October 1981.

O'Brien, Thomas C. "What's Basic? A Constructivist View," Basic Skills, Issues and Choices, first of two books, April 1982.

O'Brien Thomas C. "Software of the Second-And-A-Half Kind," Classroom Computer Learning, September 1983.

O'Brien, Thomas C. "Five Essays on Computers in Education," Phi Delta Kappan, October 1983.

O'Brien, Thomas C. "Wasting New Technology on the Same Old Curriculum", Classroom Computing Learning, November 1983.

O'Brien, Thomas C. "Computers and Education," The American Biology Teacher, Volume 47, No. 5, May 1985.

O'Brien, Thomas C. "Turning Mysteries into Problems: Software for Developing Thinking Skills," The Computing Teacher, March 1986.

O'Brien, Thomas C. "Three Anchors and Computer-Assisted Learning," National Association of Secondary School Principals Bulletin, April 1986.

O'Brien, Thomas C. "Teachers Talk -- The Failure of Computers in Education?" in press.

NOTE 3:

A Nation at Risk: Imperative for educational reform. A report to the Nation and the Secretary of Education, United States Department of Education by the National Commission on Excellence in Education, April 1983.

NOTE 4:

Callahan, Raymond E. Education and the Cult of Efficiency, The University of Chicago Press, 1962.

NOTE 5:

Labinowicz, Ed The Piaget Primer: Thinking, Learning, Teaching, Addison-Wesley Publishing Co., 1980.

Labinowicz, Ed Learning From Children, Addison-Wesley Publishing Co., 1985.

NOTE 6:

SEEDBED (a journal by teachers for teachers) is a quarterly publication of the Teachers' Center Project of Southern Illinois University of Edwardsville.

NOTE 7:

Hunt, J. McV. "The Impact and Limitations of the Giant of Developmental Psychology," Studies in Cognitive Development Elkind, David and Flavell, John H., Editors, Oxford University Press: New York, 1969, pp. 3-66.

NOTE 8:

A "software index" in Classroom Computing Learning, May 1986, describes more than 150 "noteworthy" software titles.

The 1986 Educational Software Preview Guide, developed by the Educational Software Evaluation Consortium and published in the May 1986 CUE (Computer Using Educators) newsletter lists almost 1,000 "favorably reviewed" microcomputer software pieces, 155 of them in mathematics, 72 in problem solving and logic, and another 67 in science. Copies of the report are available from the California TECC Software Listing and Clearing House, San Mateo County Office of Education, 333 Main Street, Redwood City, CA 94063.

Another superb list of software recommendations is "Selecting Computer Software -- We Take It Seriously!" by Jean Donham (Iowa City Schools) in the October 1984 issue of The Computing Teacher. An extension of the list is currently being written, I am told, by Jean Donham and Robert M. Van Deusen (Grante Wood AEA, Cedar Rapids, IA).

CLASSIFICATION UPDATE - DEWEY 004-006

By: Charlene Orr, Maine Township High School East, Park Ridge, IL

004 DATA PROCESSING COMPUTER SCIENCE

Use 004 for generalities of data processing and computer science; selection and use of computer hardware and computer systems; and processing modes.

Use 621.39 for the engineering or manufacturing of computer hardware.

.015 1 Computer mathematics [formerly 519.4]

.019 Human-computer interaction
Include comprehensive works on psychological and non-psychological aspects of human-computer interaction

.11 Digital supercomputers

.12 Digital mainframe computers

.125 Specific digital mainframe computers

.14 Digital minicomputers

.145 Specific digital minicomputers

.16 Digital microcomputers
Personal computers, comprehensive works on minicomputers and microcomputers

.165 Specific digital microcomputers

.19 Hybrid and analogue computers

.2 Systems analysis and design, computer architecture, performance evaluation

.21 Systems analysis and design
Analysis of a user's problem preparatory to developing a computer system to solve it

.22 Computer architecture
The design and structure of the computer - and the computer in relation to its peripheral devices

.24 Performance evaluation
Performance measurement and evaluation to aid in designing or improving the performance of a computer system

.3	Processing modes
004.5	Storage
.53	Internal storage (e.g., random-access memory, RAM and read-only memory, ROM)
.56	External storage Hard and floppy disks; tapes, cartridges, cassettes; tape and disk drives; optical storage devices; punched cards
.6	Interfacing and communication Equipment and techniques linking computers to peripheral devices or to other computers
.64	Modems, optical-fiber cable, etc.: various hardware for interfacing
.7	Peripherals
.76	Input peripherals : card readers, keyboards
.77	Output peripherals : printers, monitors
005	COMPUTER PROGRAMMING, PROGRAMS, DATA
.1	Programming (writing instructions to direct the operation of a computer or its peripheral equipment).
.13	Programming languages
.131	Symbolic (mathematical) logic
.133	Specific programming languages
.136	Machine and assembly languages
.14	Programs that aid in debugging other programs
.15	Preparation of program documentation Class program documentation with the program in 005.3 Class effective technical writing in 808.06 (paragraph structure, organization for clarity, use of active voice)
.2	Programming for specific types of computers
.26	Programming digital microcomputers

.3 PROGRAMS
Software and programs on microchips (firmware)
General concepts: history, description, critical appraisal, selection, use
Application programs (programs designed to teach, entertain, or perform other tasks for users) and software documentation

.329 Specific programs
Class here programs having interdisciplinary applications, such as electronic spreadsheets (useful in research, business, personal finance - anytime a matrix format is needed)

.36 Specific programs for digital microcomputers

PLEASE NOTE:

I. Class computer software by subject content

II. Multiprogram files (items that have multiple topics)
: class with the topic that receives the most emphasis

A. Emphasis may be a matter of application of one subject to another

e.g., the application of computer technology to writing instruction will be classed with writing instruction

B. For 2 subjects: class the file with the subject listed first when there is no emphasis or preponderance of either one

e.g., a file with programs that deal equally with football (796) and swimming (797) would be classed in 796, the first-listed topic

C. For several topics that are part of a broader topic:

e.g., drawing, decorative lettering, needlework, and glasswork would be classed in 740 since all of the topics are part of the broader topic

D. For several topics that are not part of a broader topic:

Use the 080s (General collections)
030s (General encyclopedic works) or
050s (General periodicals), or choose
005.3 (Collections of programs, systems of interrelated programs, individual programs having interdisciplinary application)

005.43 SYSTEMS PROGRAMS OPERATING SYSTEMS

Utility programs: systems programs that perform housekeeping tasks. They do not directly perform user's tasks, but enable a computer and its peripherals to operate so that the user's programs will work.

e.g., ProDOS
M/S DOS
CPM
Apple DOS

005.45 Programming - language translators
Programs that allow programmers to write programs in a language closer to natural language than a machine can understand. Translators convert programs written in a programming language into a language acceptable by a machine.

005.6 Microprograms and microprogramming
Does not mean programming or a program for a microcomputer. Microcomputing = writing programs in which each instruction specifies a minute operation of the computer.

.7 Data in computer systems

.74 Data base management systems

.75 Specific types of data files and databases

.8 Data security (Access control)

006 Special computer methods
Programs, programming, selection and use of of hardware for special computer methods such as artificial intelligence and computer graphics

.3 Artificial intelligence (formerly 001.535)
Use of computers for solution of complex problems involving reasoning; study of computer techniques to supplement the intellectual capabilities of human beings.

.4 Pattern recognition (Includes bar-code scanning and and scanners)

.5 Sound synthesis (use 789.99 for computer music)
A sound synthesizer, attached to the computer, acts on signals from the computer and can put sounds together to make words as well as music

.6 Computer graphics
Class here comprehensive works on computer graphics and computer sound synthesis.

.62 Hardware
e.g., graphics terminals, plotters. Class works that treat equally the use of equipment for graphics and non-graphics tasks in 004.

.66 Programming computer graphics (Including languages)

.68 Programs

MISCELLANEOUS TOPICS AND THEIR NUMBERS

Computer-aided manufacture	670.427
Computer art	760
Computer communications	384.3
Computer games	
A. Class educational games with the appropriate subject, e.g.:	
Math games	513
Geography of United States	917.3
B. Games designed for entertainment, e.g.:	
Electronic games	794.8
Computer games (e.g., Pac-man)	.82
Athletic and outdoor sports and games: e.g.: computerized baseball 794.86357	.86-89
C. When in doubt if a game is educational or recreational, use 794.8s	
Computer music	789.99
Computer operations in office services	651.8
Data processing in management	658.05
Engineering, manufacturing of computers	621.39
Financial and computer crimes	364.168
Information storage and retrieval	025.04
Word processing	652.5

CATALOGING AND CLASSIFICATION GUIDES

Beall, Julianne 004-006 Data Processing and Computer Science and Changes in Related Disciplines. Albany, N.Y. : Forest Press, 1985.

Comaromi, John P., ed. Manual on the Use of the Dewey Decimal Classification: Edition 19. Albany, N.Y. : Forest Press, 1982.

Dodd, Sue A. Cataloging Microcomputer Files : a Manual of Interpretation for AACR2. Chicago, IL : ALA, 1985.

Guidelines for Using AACR2 Chapter 9 for Cataloging Microcomputer Software. Chicago, IL : ALA, 1984.

Maxwell, Margaret F. Handbook for AACR2. Chicago, IL : ALA, 1980.

HOW TO RUN A COMPUTER PROBLEM SOLVING CONTEST

By: Donald T. Piele, University of Wisconsin-Parkside, Kenosha, WI

1. OPENING REMARKS

If you have made your presentation twice as concrete as you think you should, you have made it half as concrete as you ought to.

Ralph Boas - Northwestern University

2. THE INTERNATIONAL COMPUTER PROBLEM SOLVING CONTEST (ICPSC)

2.1. A definition of a computer problem solving contest.
A contest that challenges teams (1 to N students each) to solve X problems within Y hours using one computer and any language.

2.2. A brief history of the ICPSC
Ten years ago our local computer problem-solving contest was an afterthought--a diversion to keep the pre-college crowd busy while their parents and teachers attended the first annual computer fair at the University of Wisconsin-Parkside. Since we had more students signed up than we had computer terminals, we decided to allow (and later encouraged) teams of one to three students. Their task? Solve five problems in two hours using one computer and any language. Our task? Create five problems of increasing difficulty to test their ability to write clear, concise, and correct computer solutions.

Six years ago we opened the contest to educators around the world by creating the International Computer Problem Solving Contest. We supported each local site with materials, including the rules, a set of five problems and solutions for each of the three divisions: Elementary (grades 4-6), Junior (grades 7-9), and Senior (grades 10-12). Each site conducted its own local contest on the appointed day--the last Saturday in April. Local Judges graded their teams using our solutions and scoring procedure. All arrangements for recognition (awards, certificates, even pins) were supplied at the local level on a voluntary basis. Our job was to plant the seed, theirs was to make it grow.

We had an additional task--to recognize the truly outstanding teams that competed from all contest sites. Any team that solved four or five problems (this year we revised this to all five) forwarded their solutions to us for international ranking. We re-graded all entries and selected a winning team in each division, followed by second, third, and so on, until we had ranked them all. A traveling silver trophy with the engraved names of the members was sent to the winning team in each division.

The Local Contests
The contest works, I believe, because contest directors--who are local teachers or administrators--make it happen. They are responsible for recruiting judges, monitors, and contest duplicators (a person not involved with training students, who duplicates our problems and keeps them secure until the day of the contest). I

know the work involved in running a local contest first hand through our local contest at UW-Parkside and through the many letters we receive each year from the other contest sites. This year, for example, Joan Murray wrote from Royal West Academy in Montreal, Canada and sent along a wonderfully detailed account of their well-organized local contest along with a lapel pain, beautifully shaped in the form of a floppy disk, which is given out with a certificate to those who participate.

Another interesting letter arrived from Betsy Janssen of the Albuquerque, New Mexico public schools who ran a contest for 39 teams from 21 local schools. She wrote:

"We have been working hard to move away from teaching programming languages as an end goal and towards a focus on learning to solve problems. I think this year's contest preparation and contest itself did more to further those goals than all of our syllabus writing and rewriting... We really appreciate being able to participate in the ICPSC. Getting the logistics sorted out is a big job. It would be a nearly impossible task to do both the logistics and inventing interesting, appropriate problems."

Someone asked me recently, "What keeps a team from being dishonest and finding out the problems ahead of time?"

It certainly is a possibility, especially in today's world of instant telecommunications. In fact, this year a team entering our own local contest had knowledge of the problems through another local contest site in Illinois held on Friday, April 25th, the alternate date. When apprised of the situation, we were able to correct it, but what about all the other sites?

My first response was, "With so little at stake, why would anyone bother? Our first prize is only a traveling trophy."

Nevertheless, we always take precautions to be sure that the top ranked team in each division is truly deserving. We phone each winning contest director or coach to confirm our judgment.

2.3. The 1987 ICPSC

Probably no two contest sites run their local ICPSC the same way. That's because we don't require that they do. But there are a few things we do require.

Contest Director

The person responsible for organizing the local ICPSC site is the Contest Director.

Contest Duplicator

Each contest director must appoint a contest duplicator. This person is responsible for receiving, duplicating, and keeping secure the new set of contest problems. We send one copy, and the duplicator makes as many duplicates as necessary. During the contest, each team should have its own set of problems. This person must not be involved with the training of any students involved in the contest. A secretary is a typical choice.

Judging

The first round of judging is done at the local contest site by local judges chosen by the contest director. We provide the forms, grading procedure, and sample solutions to assist the judges in determining the local winners. We recognize that in

some situations our procedures may not be the best. Therefore, a local contest site is free to substitute its own grading procedure, so long as it determines which teams, if any, solved all five problems correctly.

We provide the second round of grading by comparing the results from only those teams that solve all 5 problems correctly. We look for solutions that solve the problem as stated, are easy to read, are short, fast, and creative.

Awards

All local awards are the responsibility of the local contest site. International recognition is provided by us. All individuals on teams that we judge to have solved all five problems correctly will receive a certificate of achievement. The top ranked team in each division receives a silver traveling trophy with the winners' names engraved on it.

Problem-Solving Emphasis

Teachers of Mathematics identified the improvement of problem-solving skills as its primary objective for the 1980's. The ICPSC supports this emphasis by challenging pre-college students to combine their programming skills with problem-solving. We reward creative, simple, short, and clear computer algorithms to five kinds of problems: 1) computation, 2) simulation, 3) graphic patterns, 4) word, and 5) mind bender. Any computer and any programming language can be used. We actively support the language BASIC, Pascal, and Logo.

New Elementary Logo Contest

The type of problems that one can solve with BASIC do not overlap well with the kinds of problems suited for Logo. It it practically impossible to create a set of problems for elementary students that can be as easily solved in Logo as BASIC, and vice versa. Since many elementary schools teach Logo and not BASIC, we have decided to try a separate Logo Division at the elementary level--grades 4-6.

There are some difficulties, however, as Sharon Burrowes pointed out to me recently. The biggest hassle is printing pictures. Some versions of Logo allow it while others do not. For example, Terrapin Logo allows it and Apple Logo does not. Even the printing of text varies from version to version. The problem with all this comes in judging each teams' work. If the contest site is small, this inconvenience can be overcome locally. The contest judges could decide to grade the Logo programs by having them submitted on a disk and running them. We would leave this decision up to each local contest site. In fact, this may turn out to be the best way to judge the Logo solutions at the international level. If you have any ideas, let's hear from you. We are still working out the best format for judging the Logo Division.

2.4. The 1987 Rules

DIVISIONS:

There are four divisions in the ICPSC:

Senior (SR) Grades 10-12 Age <=18
Junior (JR) Grades 7-9 Age <=15
Elementary (EL) Grades 4-6 Age <=12
Logo Grades 4-6 Age <=12

Students enter a division as a team of one, two, or three students. All members of the team must be in the required grade range or lower to be entered in a division. (The ages are listed above to assist schools outside of the United States establish grade equivalents. In the U.S., use the grade level to determine the

appropriate division.)

COMPUTER SYSTEMS:
Any computer system and computer language (for the EL, JR, and SR divisions) may be used; however, each team may use only one input device (keyboard/terminal). If a team chooses to use a compiled language, such as Pascal, they may use a separate computer for compiling purposes only. A printer must be used for listing the programs and displaying all sample runs. All scoring is done from hard copy. (If necessary, local judges may disregard this hard copy rule for the Logo Division.)

TIME LIMITS:
The time limit is exactly two hours. Each team attempts to solve as many problems as possible from a set of five problems and to save them to disk. After the two-hour time period, each team may take up to 30 minutes to list its programs and print its sample runs on paper. No changes may be made at this time to the programs without the permission of the judges. For example, judges may grant permission to make changes in the PRINT and INPUT statements that may be necessary to make a printed copy of the sample run. Any program that cannot print a sample run within 5 minutes is disqualified as being too slow.

SAMPLE RUNS:
All sample runs that are asked for in the problem must be completed. They must accompany the program listing. The listing and all sample runs must be stapled to a PROGRAM SCORE CARD -- one for each problem submitted. Do not use screen formatting commands that require screen dumps to print a sample run. Screen dumps are not valid sample runs. (This rule does not apply for the Logo Division.)

GRADING PROCEDURE:
The best solutions have these three properties: (a) they are correct and solve the problem as stated; (b) they are well-structured and easy-to-read; (c) they are short, sweet, and creative. Each solution is worth at most 20 points. The points are broken down as follows:

Correct Solution: Fifteen (15) points for each program that runs correctly. One to five (1 to 5) additional points for design and ease of reading.

Good Try: Seven (7) points for each program that almost runs correctly or is a solution to a slightly different problem. One to three (1 to 3) additional points for design and ease of reading.

We Tried: Four (4) points for an attempted solution. One (1) point for any and all sample runs.

General:
No outside help is allowed during the contest, including books, programs, or students not on the team. The only material allowed is a list of commands for the language or the printer. If you have any questions, ask a local judge.

2.5 Ten Steps to Running A Local Contest

Making the arrangements for a local contest involves a number of steps. Here is the sequence of steps that we follow at the University of Wisconsin-Parkside.

Step 1
We send our invitations to schools in our area to enter teams of 1 to 3 students each. They fill out a registration form and reserve a computer. We have enough computers at the University to run the contest, but any team may bring their own if they choose. If they are new to the contest, we provide them with a copy of previous problems.

Step 2
On the day of the contest, we set up a registration table to check teams in. A set of INSTRUCTIONS TO THE TEAMS is handed out for each team to read.

Step 3
Ten minutes before the contest, the teams are gathered together for final instructions. At this time one set of problems is placed at each station.

Step 4
The contest begins. Judges circulate through the computer area answering questions and monitoring the teams. Two hours later, the teams are instructed to stop programming and begin printing the programs and sample runs. Each sample run must have a program score card attached to it, and the set of programs from each team must have a master score card attached to it. We assign a number to each team. We recommend that teams print the division, problem number, and team number in the heading of each program listing.

Step 5
The judging begins. We split up the entries by problem number and assign one judge to each problem. After the judging has been completed, we gather the results together by team and record the scores on a spreadsheet. The total scores are sorted and a copy of the spreadsheet is printed out and distributed.

Step 6
While we are grading the programs, the teams have lunch and enjoy free access to the University's recreation center. Two hours after the programming ends, trophies are presented to the top three teams in each division, and certificates are given to the team members.

Step 7
We fill out the contest results form and prepare to submit to the ICPSC the programs of any team that solved all five problems.

Step 8
The staff helps clean up the computing area and returns all equipment to its proper place.

Step 9
We submit a list of our local winners to the University's public information office, which writes a story for the local newspapers.

Step 10
That evening, the helpers and judges are treated to dinner out at a local restaurant.

2.6 The 1990 ICPSC

The Future

Could success ruin the ICPSC? Probably, if people begin to take it too seriously.

I want the ICPSC to stay around and also to grow. This year at the annual board meeting of the International Council on Computers in Education, ICCE added their support. My next step is to seek out corporate sponsorship that can help bring our computer problem-solving challenge to the attention of more schools in the U.S. and abroad. The potential for growth is demonstrated by a letter I received this year expressing an interest in our contest from Tan Hao Qiang, Vice-President of the Beijing College of Automation Engineering and Vice-President of the National Institute of Computer Science Education. He views our contest as one way of encouraging the use of computers in China's schools.

The ICPSC is dedicated to promoting the uniquely human skill of problem solving. The computer is the tool, and the language a means of communicating with the tool. In the future, as the computers and the languages change, the fundamental challenge of the ICPSC will remain constant. Lest we forget, the words of George Polya will remind us.

"A great discovery solves a great problem but there is a grain of discovery in the solution of any problem. Your problem may be modes; but if it challenges your curiosity and brings into play your inventive faculties, and if you solve it by your own means, you may experience the tension and enjoy the triumph of discovery."

How To Solve It

2.7 Registration Form

7TH ANNUAL INTERNATIONAL
COMPUTER PROBLEM SOLVING CONTEST
1987 REGISTRATION FORM

Contest Director _____

School or Organization _____

Address _____

City, State, Zip _____

Phone (_____) _____

Contest Size: Approximately how many teams will enter your local contest?_____

Divisions (Circle the divisions you want to enter):

Registration Fees:

Logo	Elementary	Junior	Senior
Grades 4-6	Grades 4-6	Grades 7-9	Grades 10-12
$20.00	$20.00	$20.00	$20.00

(For any three, the fee is $50.00. For all four, the fee is $60.00.)

Total Fee: _____

Did you enter the contest last year? _____
Do you want a copy of the previous Senior Solutions in Pascal?_____

Contest Duplicator: (The person responsible for making copies of the contest problems and solutions. This person should not be involved in teaching or coaching students.)

Contest Duplicator: (The person responsible for making copies of the contest problems and solutions. This person should not be involved in teaching or coaching students.)

Contest Duplicator: _____

School: _____

Address: _____

City/State/Zip: _____

Return by April 1 to: D. T. Piele
ICPSC
Box No. 2000
University of Wisconsin-Parkside

2.8 Support Materials

2.8.1. Books of previous contest problems and solutions

2.8.2. Basic, Logo, and Pascal Solutions.

2.9 References

"How To Solve It--With The Computer:International Computer Problem-Solving Contest." Creative Computing 7(2): 82-91, February 1981.

"How To Solve It--With The Computer: International Computer Problem-Solving Contest Results." Creative Computing 7(10): 140-148, October 1981.

"How To Solve It--With The Computer: International Computer Problem-Solving Contest." Creative Computing 8(1): 154-163, January 1982.

"How To Solve It--With The Computer: International Compute Problem-Solving Contest II." Creative Computing 8(9): 130- September 1982.

"How To Solve It--With The Computer: International Computer Problem-Solving Contest II Results." Creative Computing 8(10): 140-154, October 1982.

"How To Solve It--With The Computer: International Computer Problem-Solving Contest III." Creative Computing 9(2): 132-139, February, 1983.

"How To Solve It--With The Computer: International Computer Problem-Solving Contest III Results." Creative Computing 9(10): 188-193, October 1983.

"International Computer Problem-Solving Contest (Part I)." The Computing Teacher 11(3): 21-26, October 1983.

"International Computer Problem-Solving Contest (Part II)." The Computing Teacher 11(4): 57-62, November 1983.

"International Computer Problem-Solving Contest (Part III)." The Computing Teacher 12(1): 13-19, February 1984.

"Report on the University of Wisconsin-Parkside International Computer Problem-Solving Contest." The Computing Teacher 12(4): 16-21, December/January 1984-85.

"University of Wisconsin-Parkside Computer Problem-Solving Contest--1984 Junior & Senior Division Problems." The Computing Teacher 12(6): 41-50, March 1985.

"Practice Problems For All Levels." The Computing Teacher 13(3) 29-32, November 1985.

"They Came, They Saw, They Solved." The Computing Teacher 14(1): 21-23, August/September 1986.

"It's Elementary in the Bluegrass State." The Computing Teacher 14(2): 55-56, October 1986.

"Breaking the Language Barrier." The Computing Teacher 14(3), November 1986.

"Plato Would Be Proud." The Computing Teacher 14(4): 47-49, December 1986.

3. HANDOUT

3.1 A copy of Compute It! -- The newsletter of the ICPSC.

A CONTEST IN ANY LANGUAGE

By: Nancy A. Powell, Bloomington High School, Bloomington, IL

How do you feel about these questions?

* Do you want a way to get more positive public attention for your school or program?
* Do you want more microcomputers for your school?
* Do you want more student involvement?
* Do you want your students to have more enthusiasm?
* Do you want your students to have a desire to learn?
* Do you like to stretch students' minds and cause some higher level thinking through creativity?
* Can you handle some frustration that leads to a great sense of pride and accomplishment?

If the answer to any of these questions is yes, then this contest is for you. All teachers desire enthusiasm, eagerness to learn, and good public relations; a graphics contest seems to accomplish all of these goals. The contest is a way to involve the media and draw positive attention to the schools and the accomplishments of students.

Computer programming not only stretches the mind, but programming graphics enhances creativity and the use of mathematics. When it comes to graphics, students find an eagerness that you've never seen before. All of a sudden students not only want to obtain a particular outcome, but they want to do it with pizzazz! At the end of one semester of a BASIC programming course on TRS-80 Model III's and IV's, students are able to program at least two screens of graphics with at least one of them showing some kind of animation. Students seek out new statements like LEFT$, MID$, RIGHT$, RND, and trigonometric functions such as SIN, COS, and TAN; some also choose to learn about arrays. Sound has also been used by several programmers to bring attention to their program.

Our microcomputer club, Bits and Bytes, sponsors a contest on the last day of school before the winter vacation. A copy of the entry blank for this year's contest is included. The entry form, along with a short story, is published in the school newspaper, and the contest is open, not only to students enrolled in microcomputer programming classes, but it is also open to any student in the high school. Signs are posted in the halls and in classrooms. The students have the option of programming on any microcomputer and in any language. The students have the option of bringing in their own equipment or making the necessary arrangements with people in the school to borrow equipment for the day.

If there are a number of color computers, as well as green or black and white screens, two divisions are advised, color and non-color. We have the following computers available in our school (but students often bring their own equipment in for the day): Radio Shack Models III, IV, and 2000's; Apple IIe's and Macintosh; IBM's and compatibles; and

Commodore 64's. Programs entered must be programmed so that they continually run themselves all day.

This year's contest was held all day on December 19th in the microcomputer lab. Entry forms, a $.50 entry fee, and the programs were due on Wednesday, December 17th at 3:00 P.M. The Microcomputer Club then took over by separating disks by division and finding out how many extra computers were needed and where they would be placed. Old soup cans were wrapped in holiday paper to hold the votes and decorate the lab. The votes--pennies, nickels, dimes, and quarters were deposited in the decorated cans. The money in the cans and the entry fees were used to purchase the trophies, provide prize money, and pay for any other expenses. Tally sheets were then made up and machines were assigned to each program. A team of students came in early on the day of the contest to load the programs in the computer lab, check in equipment belonging to students, and set up for the contest. Another team of students were assigned to present the trophies, clean up the lab, turn off the computers and cover them up, and check out students' equipment at the end of the day.

The management of the contest was dependent upon the participation of the "OLD" members of the micro club (most are home from college on break at this time) or parent volunteers. We had no full-time lab assistant. These members had several jobs for the day: 1) they monitored and provided security for the microcomputers and related peripherals in the lab; 2) they counted the money collected in the cans and tallied votes for each program each hour, and 3) they ate a free pizza for lunch supplied by the present micro club for their service.

Teachers had the opportunity to bring their classes through the lab all day to vote. Some teachers even gave each student a penny with which to vote! Students are not limited to putting pennies into the cans. If a $.50 is put into the can, this amount represents 50 votes. The money goes along with the entry fee to pay for the trophies for the contest. Each year the club has been able to collect between $60-$75 just from the cans. Students encourage their friends to come in and vote--the lab is open during the passing periods, as well as during class periods. This year the computer club attached a monetary award to the trophy because of the success of the contest and to encourage other programmers in the school to participate. Each year, during the contest, the programming classes have a demonstration by the robotics classes or listen to a speaker on computers.

Participation by administrators, board members, parents and the media was definitely encouraged. Support by school personnel, parents, and the media has been tremendous in the three years that the club has had the contest. The contest shows off the results of the hard work that the students have put into their projects, brings attention to the programming class, and gives the public a look at the computer facilities that are available and what could be done with them. The students get very excited when they find out how important their programs have become and this gives students that extra pat on the back that is so necessary to promote success. At the end of this report is a list of things necessary to have ready and a list of people to invite.

VARIATIONS

There are endless variations that can be made to the contest. Some of the variations include the following:

* Instead of voting with coins, distribute one ballot to each student at the beginning of class to use.

* Instead of holding the contest in a lab, move it to a more visible area in the school, such as a multi-use area, a student center, or outside of the cafeteria.

* Use the contest to promote an open house, PTA meeting, or visitation during education week.

* Instead of centering the contest around a holiday, center it around careers or other special interests.

* Use the contest to wrap up class activities. Illustrations of student-written stories on monsters, creative graphs to summarize scientific experiments, and geometrical explorations with Logo are only a few computer graphics applications.

* Have a contest to promote school spirit by designing school T-shirts with the aid of the computer. After the contest, print the winning design with the help of a program or utility onto a printer that is equipped with a transfer ribbon or take the printed design to a local screen printing shop.

* Allow students to design their graphics with the aid of graphics designing software, such as paint programs or graphics utilities.

THINGS TO DO BEFORE THE CONTEST

Print entry forms and contact the school newspaper.
Contact "old" members or find people to monitor the lab or contest area.
Wrap cans in holiday paper.
Have extra diskettes available, if necessary.
Buy trophies and/or print certificates for winners.
Print signs to advertise the contest.
Print invitations to people mentioned below and mail or deliver them.

Administrators	Teachers and their classes
Board Members	Parents
Local Newspaper	Parent-Teacher Organizations
TV Stations	

Print tally sheets for the votes.
Assign workers to help before school to load programs and set up for the contest.
Assign workers to help after school to close the lab and clean up.

ENTRY FORM

- BITS & BYTES 2ND ANNUAL COMPUTER CHRISTMAS CARD CONTEST -

NAME:_____

GRADE LEVEL: 9TH 10TH 11TH 12TH

6th hour Teacher and Room Number:_____

Computer to be used (Type, Model, Version): _____

If other than TRS 80, can you supply it? YES NO

Enrolled in Micro-Math class? YES NO
If Yes, then what hour? _____

WHEN COMPLETED RETURN TO MRS. POWELL IN ROOM 39 OR 41
by 3:00 P.M. December 17th

Paid Entry Fee? YES NO Initials_____

INSTRUCTIONAL COMPUTING: HOW DOES YOUR DISTRICT COMPARE?

By: Ted Sanders, Educational Service Center #4, Wheaton, IL

The material contained in this presentation is the result of a two-part study. In the first part of the study, a modified Delphi technique was used to develop and evaluate a list of items that would measure the extent and quality of school district computer education programs. The study was an attempt to go beyond the typical "computer counting" studies and actually get at the essence of good instructional computing programs. The second part of the study involved a survey of eighty-three randomly selected Illinois school districts to establish norms based upon the elements identified by the Delphi panel.

DELPHI

Delphi methodology is a system for using a panel of experts without the interference that can be caused by having a panel meet in a face-to-face situation. The process involves multiple iterations of a survey given to the members of a panel without their ever knowing who the other members of the committee are.

In this study, the panelists were surveyed on three separate occasions to determine and evaluate the list of essential elements found in the Essential Elements of Instructional Computing on the next page.

SCHOOL DISTRICT SURVEYS

After determining the essential elements of a good computer education program, eighty-three randomly selected Illinois school districts were surveyed to determine norms. The results of the survey are found in the section titled Survey Results-How Do You Compare?

ESSENTIAL ELEMENTS OF INSTRUCTIONAL COMPUTING
DELPHI METHODOLOGY RESULTS

The following represents the essential elements of instructional computing programs and their relative importance as identified by the Delphi panel.

not important				moderately important				critically important		
0	1	2	3	4	5	6	7	8	9	10

5.1	Drill and practice
7.1	Tutorial
6.2	Computer managed instruction
7.6	Simulations
8.4	Teaching computer-related information skills (editing text, retrieving information)
4.5	Computer programming
5.2	Computer science
8.8	Computer awareness and literacy
7.0	Computer assisted instruction
7.5	Keyboarding
7.1	Problem Solving
8.8	Business Education department usage of computers in typing, accounting, and filing instruction
6.8	Written curriculum
8.8	Inservice program
8.5	Number of computers per student
8.9	Extent of use of existing computers
6.9	Number and types of teachers using computers
7.0	Computer coordinator on staff
7.8	Local evaluation process of computer curriculum

SURVEY RESULTS - HOW DO YOU COMPARE?

The following represents the results of a survey of 83 randomly selected Illinois schools.

1) What percent of your students regularly use microcomputers for the following functions?

Unit	Elem	HS[1]	
74	77	88	drill[2]
60	51	65	tutorial
1	.06	35	CMI
40	35	65	CAI
2	41	65	simulations

2) What percent of your students receive formal instruction in each of the following areas?

Unit	Elem	HS[1]	
38	59	53	computer awareness and literacy
26	30	26	keyboarding
14	28	19	problem solving with microcomputers

3) What percent of your high school students are enrolled in computer science/ programming classes? _____

Unit	HS
16	16

4) Please circle the grade levels for which you have a formal, written computer education curriculum:

K 1 2 3 4 5 6 7 8 9 10 11 12

Unit	Elem	HS
72	45	83[3]

5) For which of the following does your high school Business Education department offer instruction using microcomputers?

Unit	HS	
77	82%	word processing
31	65	accounting (payroll, general ledger)
14	41	database management
34	47	spreadsheets

---

[1] Illinois has three distinct types of school districts--Unit or K-12 districts, Elementary or K-8 districts, and High School or 9-12 districts.

[2] Represents the percent of school districts reporting each type of instruction.

[3] Percent of districts reporting curriculum at <u>any</u> grade level.

6) What percent of your staff have participated in each of the following types of computer education inservice activities during the last year:

_____ graduate courses
_____ institute day workshops
_____ released time workshops
_____ others

Unit	Elem	HS
85	121	83[4]

7) How many instructional microcomputers does your school district presently have? _____

Unit	Elem	HS
27	20	42[5]

8) On the average how many hours per day are your microcomputers used? _____

Unit	Elem	HS
4.6	3.8	4.8

9) What percent of your staff use microcomputers in their classrooms on a regular basis? _____

Unit	Elem	HS
15.8	34	14

10) If your school district has a computer coordinator(s), please indicate the full-time equivalency for that position (one full-time person = 1, one half-time person = .5, etc.): _____

Unit	Elem	HS
17%-.43	16%-.35	59%-.43[6]

11) Please indicate the amount of money that your district spent on your microcomputer-based instructional program last year:

_____ hardware
_____ software
_____ other

Total Expenditures:

Unit	Elem	HS
$10,553	$11,053	$32,560

4 Percent totals from the four categories.
5 Averages.
6 Percent reporting a coordinator-average FTE of the coordinator.

12) Does your school district have a group of teachers and/or administrators who have been assigned the task of developing computer education curricula? _____

Percent reporting a positive response:

Unit	Elem	HS
42	51	70

13) What percent of your staff use microcomputers for the following purposes in support of the instructional program:

Unit	Elem	HS	
8	11	12	teacher utilities programs (i.e., readability programs, graphics creation aids, word search creators, test creators, etc.)
11	7.8	14	word processing
2	2	6	database management
9	5	12	grade book programs
.5	.5	1.6	authoring languages

TOPS: A DISTRIBUTED NETWORK FOR THE MACINTOSH

By: Ted Sanders and Michael P. Warner, Educational Service Center #4, Wheaton, IL

SPECIFICATIONS

Tops, published by Centrum Systems West, is networking software that utilizes AppleTalk. The software installs as a desk accessory on the Macintosh system and allows an IBM PC or UNIX system to share the network via an interface card.

DISTRIBUTED VS. CENTRALIZED NETWORKS

Centralized networks utilize single, high capacity file servers. The advantages of this type of network are:

- Control - it is a fairly simple matter to limit access to whatever pattern is desired.
- One large capacity hard disk drive (file server) may be less expensive than several smaller ones.

The disadvantages of this type of network are:

- When the file server is down, so is everything else
- Having everyone with a need to constantly access the same hardware tends to greatly slow down processing time

Tops follows a distributed network approach (although it can be used with a single, centralized file server). Any station on the network can choose at any time to become a file server for the network. Therefore, networks with several hard disk drives can have any or all of them at any given time act as file servers for the other stations on the network. Advantages of this type of network are:

- No single piece of hardware can bring the entire network to its knees
- Flexibility is insured since the role of hard drives can be redefined whenever necessary

Disadvantages of this type of network are:

- Without careful planning, confusion can reign supreme as users will not necessarily (nor magically) know what files and applications are located on which servers
- Files are not necessarily located at the most advantageous point for every user
- The users must be aware of the choices available when it comes time to save a file

HOW DOES TOPS WORK?

Since Tops installs as a desk accessory, the network may be accessed at any time - even from within an application. The choices for a user are publish, copy, or mount.

Publishing a volume (entire disk) or folder refers to the user making a volume or folder currently on his/her desktop accessible to other users on the network. It can be published in three formats: read only, one write/many read, and many writers. Only networkable, multi-user software should be published in the many writer format. At the time that a volume or folder is published, the owner of the folder may specify a password.

Copying a file is very simple. Essentially, files on the network may be copied from a published volume or folder to any station on the network. Conversely, a user at any station on the network may copy a file into any volume or folder published to the network.

Mounting a volume or folder has the effect of convincing your Mac that the volume or folder is just another disk drive connected to your Mac. This allows you to use that folder or volume within applications.

SO WHAT'S THE CATCH - WHAT DOESN'T WORK?

The network is slow. This is probably a function of AppleTalk, which functions at regular disk access speed, rather than Tops. Therefore, using Tops in a centralized network might not be a good idea.

Since Tops stands between applications and the operating system, thereby controlling disk access, some applications do not like Tops and you have to fiddle with them a bit to get them to work. Most of these applications probably try to take short-cuts around the operating system or simply do not like the new hierarchical file system.

COMPUTERS IN INDUSTRIAL ARTS GENERAL EDUCATION

By: David Siden, Deer Path Junior High School, Lake Forest, IL

COMPUTER AIDED DRAWING

Computer aided drawing, or CAD as it is commonly called today, is an example of our technology. Technology is how we apply our scientific knowledge to develop solutions to solve problems. The early brainstormers, like Thomas Edison, Alexander Graham Bell, and Henry Ford, used their freehand and mechanical drawing skills to solve problems and develop their ideas. Employees in the electric, phone, and automotive industries are still brainstorming, but today, are using computers.

In the Deerpath industrial arts classroom, the computer is used as a drawing tool. With the Apple computer and the discover CAD software, students can create a complete set of working drawings for their industrial arts projects. CAD drawings are made by the student applying mathematical principles with their hand and eye coordination using a monitor, a mouse, and a keyboard.

TURN ON TO CAD

1) Insert discover CAD disc 1 into disc drive 1 and close door.
2) Insert discover CAD disc 2 into disc drive 2 and close door.
3) Turn on the Apple computer.
4) Turn on the monitor.
5) Make sure the Caps Lock key is depressed.

DEFINITIONS

The mouse is a tool for communicating with the computer. When you move the mouse across your desktop, a pointer moves in the same direction across the monitor screen. If you run out of room for the mouse, pick it up and set it back down where you have more room. Lifting the mouse and moving it does not affect the pointer's position.

Clicking - press and let go of the mouse button.

Dragging - hold the mouse button down and move the pointer.

The menu bar - the banner you see at the top of the screen. Hidden inside the banner are the available commands in lists. To select a command, move the pointer to one of the menu titles and hold the mouse button down. This shows a pull-down menu list. Drag the pointer towards the bottom of the screen to highlight the command you want and then release the button.

DRAWING TO SCALE

If the project is larger than the drawing paper, it must be drawn smaller then its actual size. If the project, or a detail is to small, it must be enlarged. In mechanical drawing we used a ruler to draw to scale.

With CAD we will use a grid to draw to scale. A grid is a series of evenly spaced dots. These dots are arranged in rows and columns. We will assign the distance between the dots a specific measurement, thereby making the grid work as a ruler.

TO PROGRAM THE COMPUTER TO DISPLAY AND MAKE THE GRID WORK AS A RULER:

1) Position the pointer over the word Screen. Drag the pointer down to highlight the command Draw Grid and release the button.

2) Answer the computer's question by typing .5 (make sure you press the period key first so it reads point five) and press the Return key.

With the computer, as with freehand and mechanical drawing, we will use the three different degree lines (180 degree horizontal, 90 degree vertical, and 45 degree angle) to draw our projects. There are 3 ways to draw these lines: 1) Snap Grid, 2) Entity Modifiers, or 3) eye sight. For your first experience, use the Snap Grid. This command is fast and easy to use. With the Snap Grid command, the arrow needs only to be close to a Grid Dot. All lines will snap on the Grid Dots.

TO PROGRAM THE COMPUTER TO USE THE SNAP GRID COMMAND:

1) Position the pointer over the word Screen. Drag the pointer down to highlight the command Snap Grid and release the button. (You will see this command in action when you start to draw.)

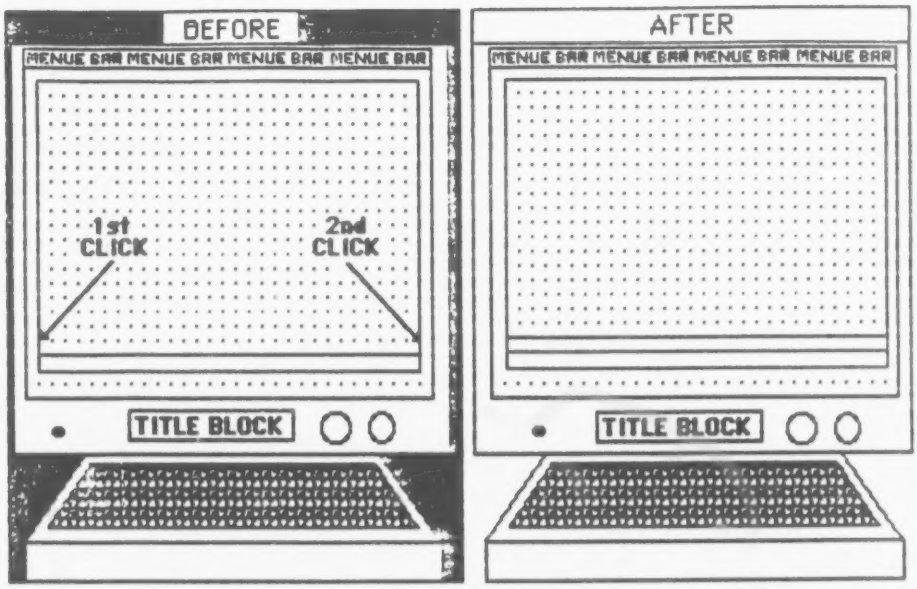
TO PROGRAM THE COMPUTER TO DRAW LINES:

1) Position the pointer over the word Entity. Drag the pointer down to highlight the command Draw Line and release the button. (The computer will draw a line between the first CLICK of the mouse and the second CLICK. With the Snap Grid command on, the pointer only has to be close to the beginning point (grid dot) and the ending point (grid dot) to draw a 180 degree horizontal line, 90 degree vertical line, or a 45 degree angle line.)

* Remember with the Snap Grid command on, the clicks only have to be close to the correct grid dot.

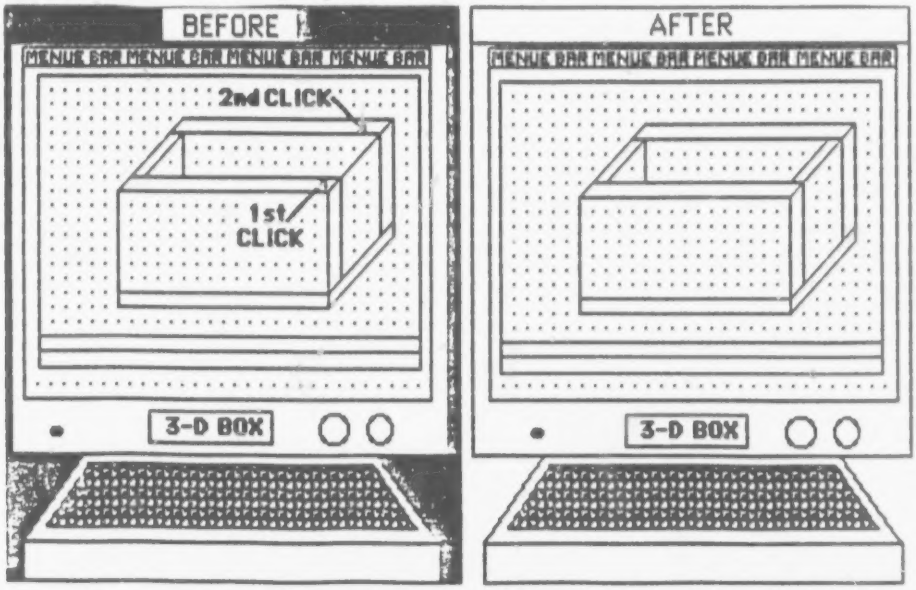
Draw border lines and a title block.
We will letter the title block after we complete a 3-D drawing.

On the following pages you will see a BEFORE and AFTER computer monitor. Notice the location and number of clicks on the BEFORE monitor. The AFTER monitor shows the results of the Clicks.



* IF YOUR LINES ARE NOT CORRECT, position the pointer over the word Entity. Drag the pointer down to highlight the command EraseLine and release the button. Position the pointer anywhere on the line and click the mouse button. The line will be removed. PLEASE REMEMBER AFTER YOU ERASE THE LINE to repeat step 9 in order to continue to draw lines.

Continue to draw a 3-D project (Pencil Box).



TO DIMENSION THE PENCIL BOX PLEASE CONTINUE WITH THE FOLLOWING STEPS

TO PROGRAM THE COMPUTER TO DRAW DIMENSIONS:

1) Position the pointer over the word Text-Dim. Press the mouse button down and drag the pointer down to highlight the command Draw Linear Dim and release the button.
2) Continue the program the computer to draw dimensions. Position the pointer over the word Modify (Text & Dimension). Press the mouse button down and drag the pointer down to highlight the command Point to Point and release the button.

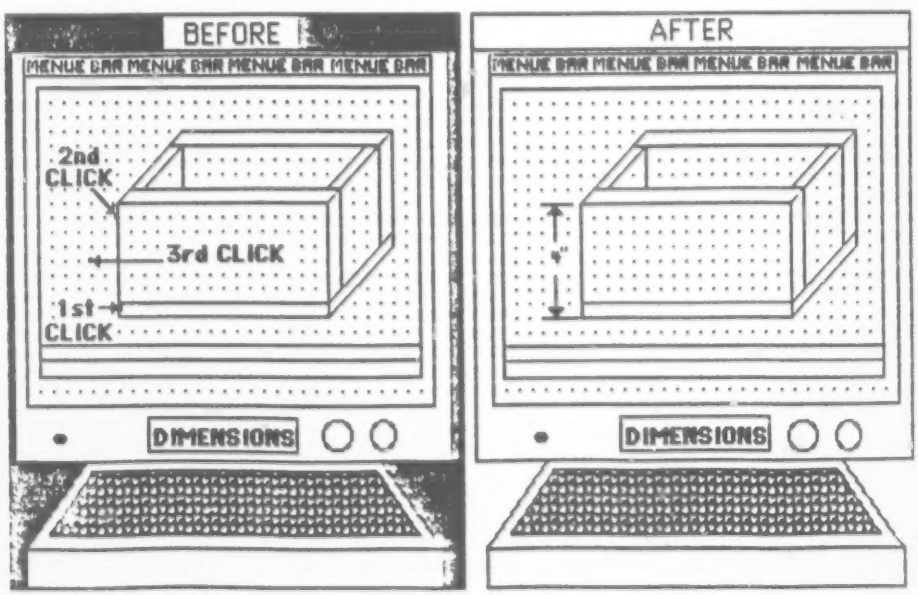
HOW TALL IS THE PROJECT?

To program the computer to draw the dimension for the height of the pencil box

1) Position the pointer over the word Modify (Text & Dimension). Press the mouse button down and drag the pointer down to highlight the command Alternat Text and release the button.
2) Type in 4"
3) Press the Control key and the Q key at the same time.

Draw the dimension for the height.

1) Position the pointer on the front left side (line 10d) near the bottom corner point and click.
2) Position the pointer on the front left side line (line 10d) near the top corner point and click.
3) Position the pointer at least one grid dot to the left and center of the left front edge (line 10d) and click.



Continue to dimension the width and length of the pencil box.

TO ADD ALL OF THE LETTERING TO THE DRAWING PLEASE CONTINUE WITH THE FOLLOWING STEPS

* The computer commands for dimensioning will automatically turn off when we program the computer with the next command.

TO PROGRAM THE COMPUTER TO DRAW THE LETTERING:

1) Position the pointer over the word Text-Dim. Press the mouse button down and drag the pointer down to highlight the command Draw Text and release the button.

Add the detail lettering (PENCIL BOX) to the front.

Because this lettering is taller than the standard size lettering, we will have to program in a modifier for the height.

1) Position the pointer over the word Modify. Press the mouse button and drag the pointer down to highlight the command Height and release the button.

2) Type 1.

3) Press the Return key.

The detail lettering is also wider than the standard size lettering.

Therefore, a modifier for width must also be programmed into the computer.

1) Position the pointer over the word Modify. Press the mouse button down and drag the pointer down to highlight the command Width and release the button.

2) Type .7.

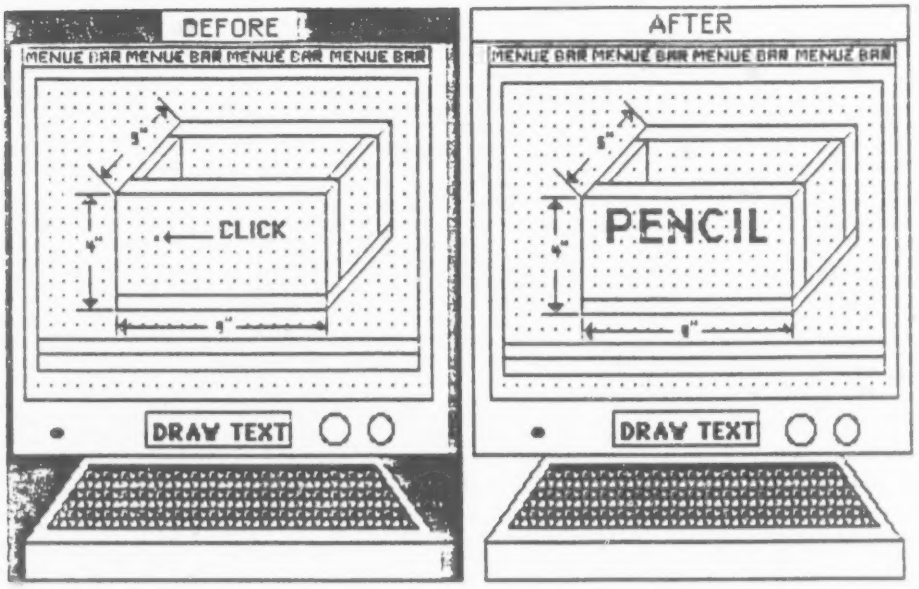
3) Press the Return key.

Add the detail lettering PENCIL to the front.

1) Position the pointer on the front, top, left corner point. DO NOT CLICK. Move the pointer 90 degrees down the left side edge 3 grid dots. DO NOT CLICK. Move the pointer 180 degrees to the right 2 grid dots and click.

2) Type PENCIL.

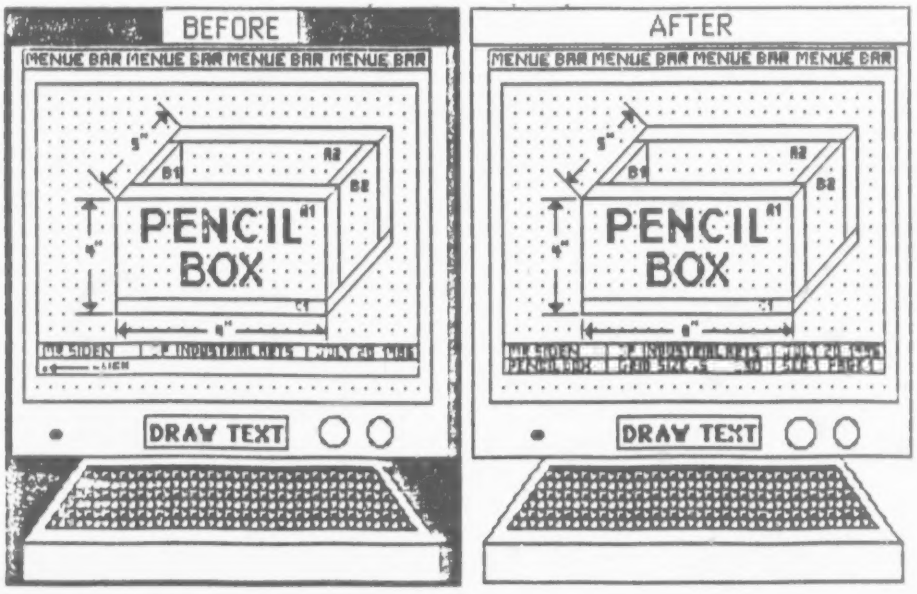
3) Press the Control key and the Q key at the same time.



Continue to draw the detail lettering of the Pencil Box

TO PROGRAM THE COMPUTER TO ADD THE STANDARD SIZE TEXT.

1) Position the pointer over the word Modify. Drag the pointer down to highlight the command None and release the button. (This command will change the text size back to the standard size).
2) Position the pointer near the bottom left corner of the first letter. Click the mouse button.
3) Type text.
4) Press the Control key and the Q key at the same time.



EXPLOITING THE STRENGTHS OF THE COMPUTER FOR TEACHING MATH AND READING

By: Martin A. Siegel, Computer-Based Education Research Laboratory, University of Illinois at Urbana-Champaign

COMPUTER IMAGINATION

Increasingly, courseware evaluation guidelines have become concerned with a quality often referred to as "imaginativeness" in tutor-style lessons. For example, one list of CBE lesson evaluation criteria includes a section called "Creativity," that asks, among other things, whether the lesson presents "information or instructional activities in a way not easily duplicated with textbooks...or other traditional classroom materials." Another checklist proposes three categories to include in courseware review: "(1) suitability of the lesson for the intended population; (2) quality of implementation of unique features of CAI; and (3) observations of student users." And a third list reminds readers:

> It is important to differentiate between the two kinds of attributes that need to be considered: (1) those that are generic to all media of instruction, and are recommended strategies to use for instructional design; and (2) those that are necessary to consider specifically in the design of software for the microcomputer and potentially affect learning outcomes in a unique way."

The characteristic of courseware that these checklists are suggesting is something I call computer imagination (Siegel and Davis, 1986). It is the degree to which a lesson exploits the strengths of the computer medium to enhance instruction, rather than merely imitating other teaching media in a computer environment. It is the most important quality of computer-based instruction and the hardest to describe in precise terms for evaluation purposes. You may find an imaginative use of the medium literally anywhere in a lesson, so to try to generate a definitive list of computer imaginative lesson traits would be very difficult.

There are many ways for a computer-based lesson to be merely imaginative--in the sense of being attractive, engaging, and "cute." But computer imagination always directly enhances a lesson's instructional effectiveness; the lesson need not contain the most dazzling electronic wizardry. A computer imaginative lesson is likely to:

1. Individualize instruction
   a. It should behave differently for different students.
   b. It should be adaptive.

2. Contain meaningful interaction
   a. It should accept student input in machine-appropriate modes.
   b. It should have immediate, instructional or intrinsic feedback.

3. Guarantee mastery
   a. It should place students accurately in instruction.
   b. It should remediate students, addressing their problems specifically.
   c. It should provide a sound review of knowledge and skills to ensure long-term memory.

A SIMPLE EXAMPLE: THE FLASH CARD DRILL

Let us examine computer imagination from the perspective of a common example, namely, the flash card drill. Classroom teachers use flash card drills as a way to help students memorize paired associate items, such as foreign language vocabulary lists. In these drills, an item is presented to the students for a response. If the response is correct, the teacher assumes the item is known and it is removed from the drill set. If the response is incorrect, the item is placed at the end of the stack to be reviewed later (see Figure 1).

Figure 1. Flash Card Drill

This format translated easily to the computer medium and was a small improvement. Drills could be implemented for individual students without requiring individual attention from the teacher. Even though such CBE drills were effective, they were not computer imaginative. They were limited by the original sequencing routines used by human teachers. A reexamination of the computer's ability to present information in any order revealed an improved sequencing strategy.

Rather than placing missed items at the end of the list, the computer can place them at pre-specified intervals and gradually increase the amount of time between each presentation--resulting in an increasing ratio review pattern that is different for each student since it is based on each student 's pattern of errors (see Figure 2).

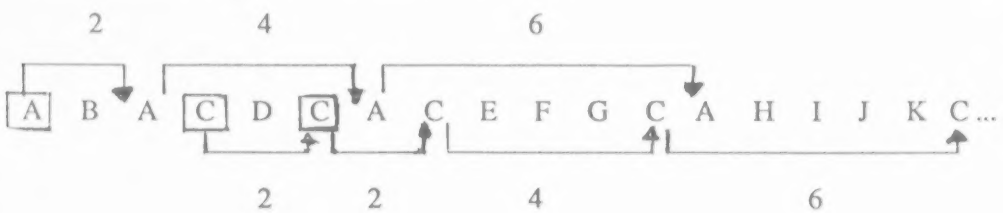


Figure 2. Corrective Feedback Paradigm

Such an approach to drills greatly enhances the efficiency and effectiveness of computer-based instruction. This strategy, the Corrective Feedback Paradigm (CFP) (Siegel and Misselt, 1984), is an instance of computer-imagination--using the particular strengths of the computer to do something that would be impossible for a teacher to do.

EXTENDING CFP TO READING

A student who responds "S" (instead of "Ag") to the stimulus "Silver" in a simple drill on the chemical elements merely needs more practice on that item. But what happens if a student, having read a passage in a lesson on WH-questions from a reading comprehension curriculum, responds to a question about when something happened with an answer that tells where? It is insufficient merely to remediate the error and then bring back the same item in review, because the object of the lesson is not to teach a specific answer to a specific question, but to teach a generalization about WH-questions.

A specific example by Siegel, DiBello, Bruner, and Felty (1977) from a reading comprehension lesson that teaches this more complex generalization is presented. The student first reads a passage like this one:

> The knitting machine was invented in England in 1589,
> and the cotton gin was invented in America in 1793.

When the student signifies that he has finished reading, the lesson presents a WH-question: Underline the part that tells when the knitting machine was invented. Two kinds of errors are possible: a discrimination error (for example, confusing when and where parts of the sentence) and other errors (for example, underlining "was invented"). Regardless of the error, the sentence on which the student made the mistake is brought back for review several times during the lesson.

Remember, though, the goal is not to teach the item itself, as in a chemical elements drill, but the generalized skill of finding the parts that answer WH-questions in any sentence. What would happen, then, if the lesson brought back the item for review just as it was presented above? Would it not be likely that the student would learn this misrule: "The answer to this sentence is 'in 1589.' That's what I should underline each time"? In other words, no generalization to the skill of finding any "when" question would have taken place. That is the limitation of normal drills.

Instead, this lesson modifies the original passage when it comes up for review. If the student failed to answer the "when" question in the item correctly the first time, the review presentation will change the original item in three increasingly generalized ways:

1. By changing the original "when" part: "in 1589" is replaced by one of these phrases:

   in the sixteenth century
   in the fifteen hundreds
   almost 400 years ago
   in the 1580s

2. Later by changing the other "when" part: "in 1793" is replaced by one of these phrases:

   soon after the Revolutionary War
   in the late 1700s
   in the eighteenth century
   shortly before 1800

3. Finally, by varying the sentence structure along with the "when" parts. For example, the order of parts within the clauses might be changed like this:

   Almost 400 years ago the knitting machine was invented in England,
   and shortly before 1800 the cotton gin was invented in America.

   Or, the order of clauses might be reversed:

   The cotton gin was invented soon after the Revolutionary War in America, but the knitting machine was invented in England in the sixteenth century.

Combined with the distributed practice feature of CFP drills, this "flipbook" technique exploits the computer medium to teach the necessary generalizations. Computer imagination in this case provides adaptive instruction and fosters conceptual understanding.

AN EXAMPLE FROM MATH

Consider how a good textbook introduces the concept of fractions to elementary students. Here is a typical procedure:

1. Start with a sheet of paper (a unit). Fold it once to illustrate the concept of "half." Fold the halves to illustrate the concept of "quarters." Demonstrate that "two quarters" is the same as "one half." Extend the concept to fractions generally.

2. Draw a number line marked off in units. Transfer the concept of folding the piece of paper into halves to the number line--dividing a unit into halves. Continue to expand the transfer by adding quarters and eighths to the divided unit.

The difficulty in teaching the concept is that it depends on students' making a critical minimum difference discrimination. A thing (a unit) becomes two halves, four quarters, etc. The trick to teaching it effectively, then, is to find a way to present the discrimination dynamically. The textbook's initial presentation of the concept relies heavily on teacher action, because books are not a dynamic medium. The problem arises when students are asked to practice the skill, applying it to other situations. The teacher must bow out and let students do it on their own, but when the teacher leaves, so does the dynamic element of instruction.

Now compare this classroom technique with the computer-imaginative lesson, "Darts," originally designed by Dugdale, Kibbey, and Cohen (1975). The computer lesson is simplicity itself: it merely instructs the student to "shoot a dart" at any point of his choosing on the number line. The lesson accepts whole numbers, fractions, and decimal equivalents (and, thus, extends its usefulness by providing a powerful means to extend the concept of fractions to decimals). The lesson retains the full dynamic quality of the teacher's presentation, but it lets the student work completely on his own. A shot that misses a balloon is as instructive as a shot that hits a balloon. Students can "zero in" on the balloons, presenting themselves with as many minimum difference discriminations as they need. Unlike the textbook exercises, their feedback is immediate. Figure 3 shows a dart that has hit a balloon at 1 2/5, and shows a dart that has not hit a balloon.

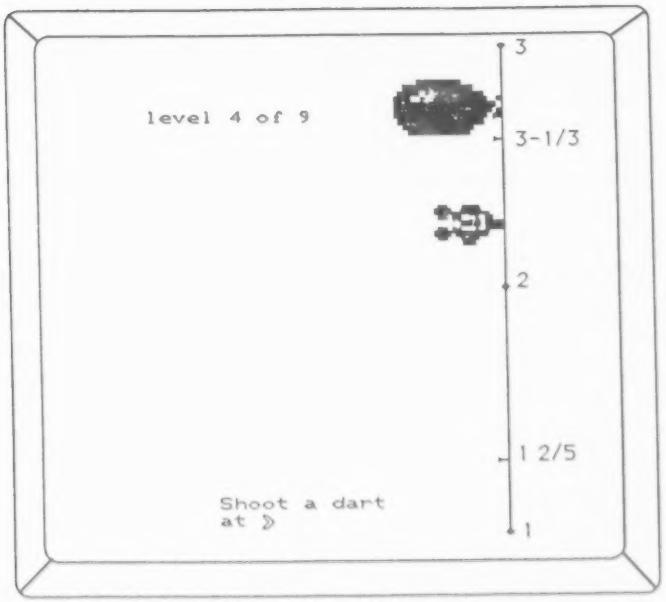


Figure 3. Darts

The basic design of the lesson is computer imaginative enough, but its full instructional power becomes apparent when you begin to consider the ways students use it. When zeroing in on a target balloon, for example, they can learn important things about mathematical expressions. Looking at the number line, for example, a student may realize that the balloon is "almost to 3 but not quite." He may instruct the program to shoot a dart at "2+2/3," but he may also try "3-1/3." If that does not quite do it, students sometimes type instructions like "3-1/3+.1" to produce the effect they desire.

Dugdale and her colleagues also report that students frequently locate the position of a balloon by doing things like measuring the total number of finger widths between two integers marked on the number line, and then shooting a dart at a fraction expressed in that measurement. For example, if the space between "0" and "1" on the number line is 4 finger widths, with a balloon near 3 finger widths, the student will shoot a dart at "3/4." If instead, the space between "0" and "1" measures 4 1/2 finger widths, the student may decide to use 9ths, calling each finger width equal to 2/9.

Such behavior can lead to insights about equivalent fractions, but, combined with another feature of the lesson, it does something even more valuable: it integrates learning about estimation and approximation with the teaching routine, instead of treating it separately in the curriculum. The "correct" answer is any answer that pops the balloon, and balloons appear on the number line at random points that are not always expressible in simple fractions with small denominators. The lesson thus teaches important practical applications related to the concept of fractions: that measurement is not exact, and that the determination of how much accuracy of measurement is necessary depends upon the task at hand.

"Darts" is an elegant, economical, and computer-imaginative lesson, but I chose this illustration to demonstrate how exploiting the strengths of the computer as an instructional medium often leads to ways of dealing with subject matter that look very different from an appropriate treatment in other media. It would have been quite possible to construct a computer lesson that let students apply the concept of fractions to new situations exactly as the textbook did. Such a lesson could even have been effective. But the treatment in "Darts" is better precisely because of its computer imaginativeness.

CONCLUSION

High quality courseware must be developed with great care to avoid the problems that plague much of the available instructional software. The key to designing excellent courseware is computer imagination , or exploiting the particular strengths of the computer as an instructional medium (Siegel and Davis, 1986). Appropriate uses of computers are not those that mimic other established methods, but rather those that uniquely employ the computer's capacity for extensive data manipulation and storage, immediate interaction, graphics, and calculation. In the use of computers, it is important to realize that we are not so much interested in computers as in the content that flows through them. The more transparent the medium (that is, the easier it is to deal directly with the content and not the computer), the more benefit the student will gain from the computer interaction.

REFERENCES

Dugdale, S., Kibbey, D., and Cohen, B. (1975) "Darts," Computer-based education lesson.
Urbana: University of Illinois.

Siegel, M. A. and David, D. M. (1986) Understanding computer-based education.
New York: Random House.

Siegel, M. A., DiBello, L. V., Bruner, E. C., and Felty, J. M. (1977)
"PCP reading comprehension I-IV," Computer-based education lessons.
Urbana: University of Illinois.

Siegel, M. A. and Misselt, A. L. (1984) "An adaptive feedback and review paradigm for computer-based drills," Journal of educational psychology, 76, no. 2, 310-317.

"DIRECT ENGAGEMENT" SOFTWARE FOR TEACHING MATHEMATICS

By: Patrick W. Thompson, Illinois State University, Normal, IL

We all know that computers can have a positive effect on the quality of students' learning and the quality of teachers' teaching. Or do we? Whether or not computers have a positive effect on teaching and learning depends upon what we take as good learning and good teaching. National and international surveys of mathematical performance suggest that even though today's students may be skilled in applying procedures, they are, quite largely, unskilled in *applying* those skills. Our criteria for good learning must go beyond the development of skills *per se*. Our criteria for good teaching must go beyond teaching for skills.

The focus of my research is to take advantage of the unique powers of today's computers in order to provide students with opportunities to develop powerful conceptualizations of standard mathematical content—conceptualizations that give a foundation for generalization and transfer. The approach I have taken centers around the idea that mathematical systems are fundamentally representational: Actions on symbols are reflective of actions within some represented system. Two programs—one for decimal numeration and one for elementary algebra—will be used to illustrate this concept. The design of the software is in line with what Draper and Norman (1986) call *direct engagement software.*

> One approach, a rather traditional one, is to start with considerations of the person, the study of the human information processing structures, and from this to develop the appropriate dimensions of the user interface. .... Another approach is to examine the subjective experience of the user and how it might be enhanced. When we read or watch a play or movie, we do not think of ourselves as interpreting light images. We become a part of the action: We imagine ourselves in the scenes being depicted. We have a "first person experience." So too with well-constructed video arcade games and with simulation tools. Well, why not with computers? The ideal associated with this approach is the feeling of "direct engagement," the feeling that the computer is invisible, not even there; but rather, *what is present is the world we are exploring, be that world music, art, words, business, mathematics, literature, or whatever your imagination and task provide you* [italics added].(Draper & Norman, 1986, p. 3)

THE BLOCKS PROGRAM

The program for decimal numeration, called **BLOCKS**, provides students with an environment in which quantities are represented in two systems: base-ten numerals and base-ten blocks (actually, other bases can be used as well). Anything they do to the blocks is reflected in the numeral representation of the quantity. Anything they do to the numeral representation of the quantity is reflected in the block representation. Standard algorithms for operating on numerals are presented as efficient schemes for recording actions on structured representations of quantities.

Figure 1 shows the primary display in which students work. Students create block representations of quantities by clicking on blocks and dragging the image into a region. In BLOCKS ONLY mode, they can operate on quantities only by doing things to the block representation of the quantity. The two primary operations on blocks are GLUE and

UNGLUE (Figure 2). To unglue a block, students select the block (by clicking on it) and then clicking the UNGLUE button. In other words, they say (by pointing and clicking),

"This is the block I want to act on. The action I want performed is to unglue it." To glue blocks, students select ten of the same kind (except for the cube) and then click the GLUE button. After an action on blocks is performed, the numeral representation is updated accordingly.

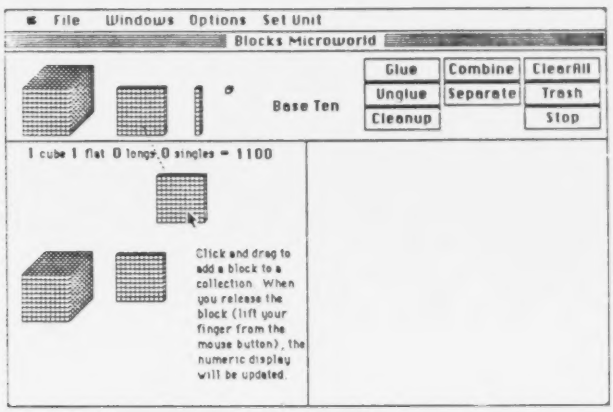

File Windows Options Set Unit
Blocks Microworld
Glue
Combine
ClearAll
Unglue
Separate
Trash
Cleanup
Stop
Base Ten
1 cube 1 flat 0 longs 0 singles = 1100
Click and drag to add a block to a collection. When you release the block (lift your finger from the mouse button), the numeric display will be updated.

FIGURE 1.

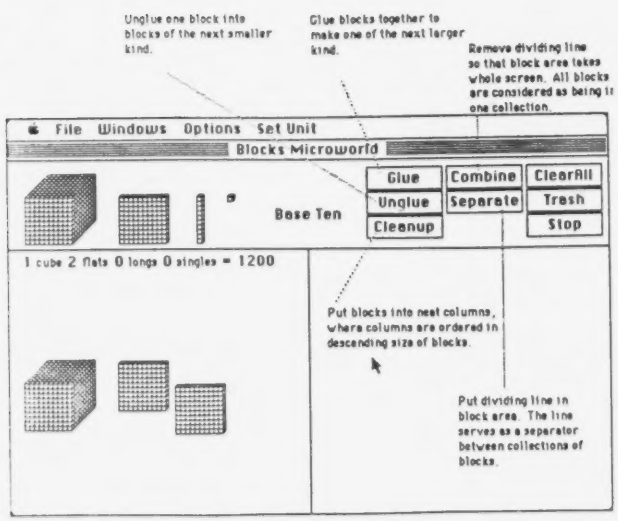

Unglue one block into blocks of the next smaller kind.
Glue blocks together to make one of the next larger kind.
Remove dividing line so that block area takes whole screen. All blocks are considered as being in one collection.
File Windows Options Set Unit
Blocks Microworld
Glue
Combine
ClearAll
Unglue
Separate
Trash
Cleanup
Stop
Base Ten
1 cube 2 flats 0 longs 0 singles = 1200
Put blocks into neat columns, where columns are ordered in descending size of blocks.
Put dividing line in block area. The line serves as a separator between collections of blocks.

FIGURE 2.

A second kind of action is to move one or more blocks from one quantity region to another. Figure 3 shows this action. With this facility, students can effect the conceptual actions of "put together" and "take away." Whenever they move blocks, the numeral displays are updated accordingly.

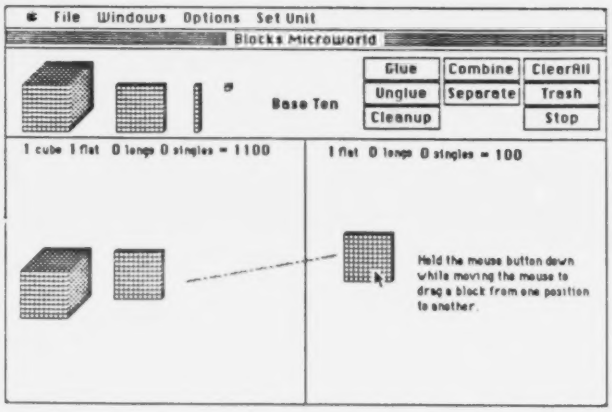


FIGURE 3.

The idea of operating in either of two representational systems is effected by allowing students to switch from operating on blocks to operating on numerals. Figure 4 shows the menu in which students make that choice, as well as some other options they can choose. Figure 5 shows the display after a student has chosen to operate on numerals. The only difference is that the buttons GLUE and UNGLUE are replaced by the buttons CARRY and BORROW. If a student selects a number of blocks and then clicks BORROW, the program will act out with blocks the meaning of borrowing—it will select a block of the appropriate size and unglue it. The numeral representation will be updated accordingly. If a student selects a number of blocks and clicks CARRY, the program will select ten blocks of the appropriate size and glue them (if there are not at least ten of that size, the program will do nothing).

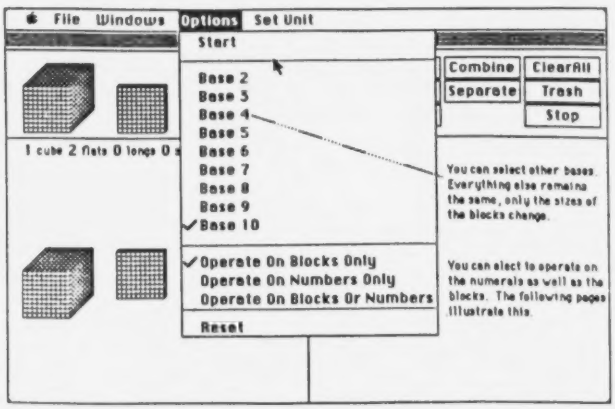


FIGURE 4.

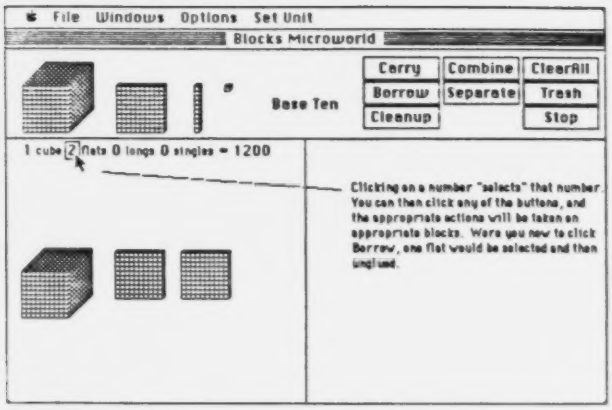

File Windows Options Set Unit
Blocks Microworld
Carry
Combine
ClearAll
Borrow
Separate
Trash
Cleanup
Stop
Base Ten
1 cube 2 flats 0 longs 0 singles = 1200
Clicking on a number "selects" that number. You can then click any of the buttons, and the appropriate actions will be taken on appropriate blocks. Were you now to click Borrow, one flat would be selected and then unglued.

FIGURE 5.

To develop concepts of decimal numeration, the program is designed so that any of the four sizes of blocks can represent one. Thus, when a student selects FLAT IS THE UNIT from the Set Units menu, the cube represents ten, the flat represents one, a long represents one-tenth, and the single represents one one-hundredth (Figure 6). However, all actions remain the same.

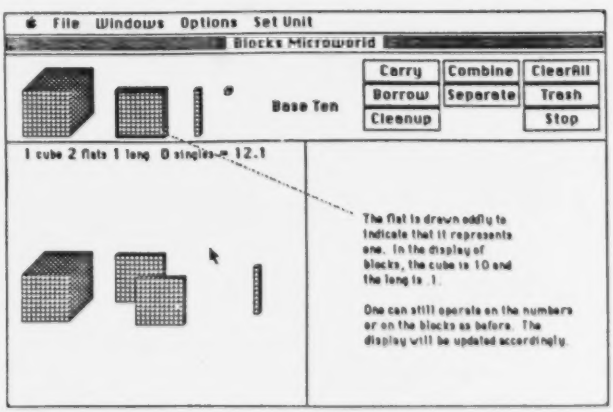

File Windows Options Set Unit
Blocks Microworld
Carry
Combine
ClearAll
Borrow
Separate
Trash
Cleanup
Stop
Base Ten
1 cube 2 flats 1 long 0 singles = 12.1
The flat is drawn oddly to indicate that it represents one. In the display of blocks, the cube is 10 and the long is .1.
One can still operate on the numbers or on the blocks as before. The display will be updated accordingly.

FIGURE 6.

The **BLOCKS** program presents a uniformly consistent model of numeration—both whole number and decimal—for any base from two to ten. That is, just as one would teach decimal numeration in base ten, one could teach decimal numeration in base six were one so inclined.

The **BLOCKS** program was designed with several objectives in mind and designed with the assumption of a particular development of the concept of algorithms.

A principal objective of **BLOCKS** is to convey the idea that quantities can be represented in a variety of ways, and each representation is more useful than others in a particular context. The blocks themselves, as representations of quantities, are useful for understanding the structure of base numeration and for developing efficient methods for adding and subtracting. Numeral representations are more efficient for skilled processing than are blocks. However, algorithms developed for either system translate directly into algorithms within the other. Moreover, standard column addition and subtraction algorithms can be presented as efficient schemes for recording the results of actions in either representational system.

Development of algorithms in the context of the **BLOCKS** program is as follows:

- Develop efficient methods for addition and subtraction in BLOCKS ONLY mode.
- Develop efficient methods for addition and subtraction in NUMBERS ONLY mode.
- Develop efficient methods for *recording* the results of actions in either mode.
- Repeat step 3, but *without* using a computer; problems posed in horizontal format.
- Repeat step 3, but *without* using a computer; problems posed in vertical format.

The **BLOCKS** program has been used to date only with prospective elementary teachers in a mathematics methods course. The students' reactions have been most encouraging. In the Fall of 1987, the **BLOCKS** program will be used with a class of seventh graders in a study of the development of their concepts of decimal numeration.

THE ALGEBRA PROGRAM

The program for teaching elementary algebra is more sophisticated than the one just described for numeration, so it will not be possible to describe it fully in the limited space allowed here.

The essence of the algebra program, called **EXPRESSIONS**, is that it presents expressions and equations in two formats. The two formats are regular (sentential) notation and expression trees (Figure 7.) The purpose of displaying expressions and equations as expression trees is that they show explicitly an expression's or equation's structure. Students act on expressions and equations by acting directly upon their structure as depicted in their expression trees. The display shown in Figure 7 came about by a student typing the expression (a+b)+(c+d) and then clicking the button labelled ASSOCL, which means "apply the associative property from left to right." Figure 8 shows the display after the student clicked the plus sign at the top of the tree, which indicated that it was the entire expression to which the associative property was to be applied.

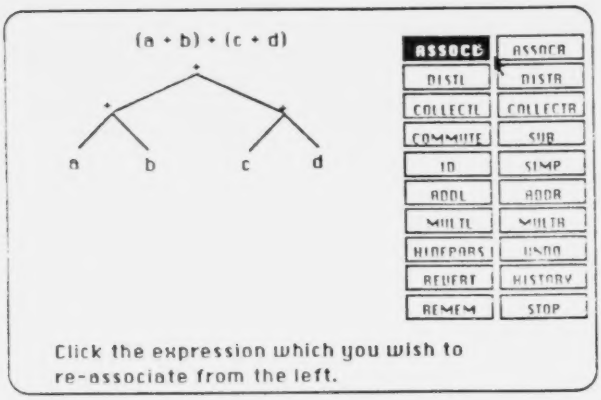


FIGURE 7.

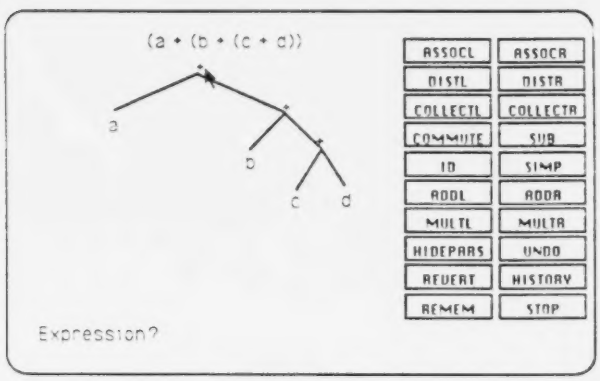


FIGURE 8.

Students can apply identities to expressions and create their own identities, by use of the ID button (Figure 9). A student selects an identity to apply to an expression or subexpression by clicking ID, selecting the identity to apply, and then clicking on the operation sign that defines the expression or subexpression to which the identity is to be applied.

Students can create identities by clicking on the REMEM button (Figure 10). This stores the original expression and the current expression as an equivalent pair and adds that pair to the list of identities. The new identity can be used in the future just as can those that were already in the list.

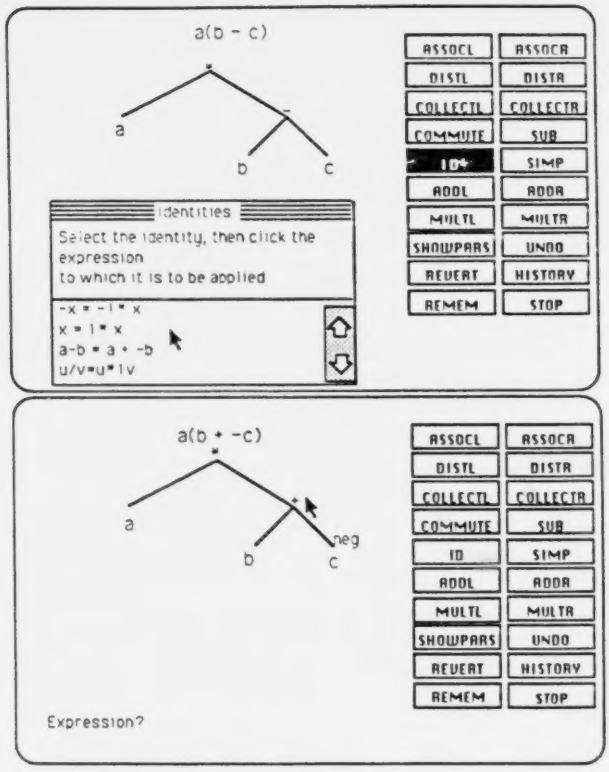


FIGURE 9.

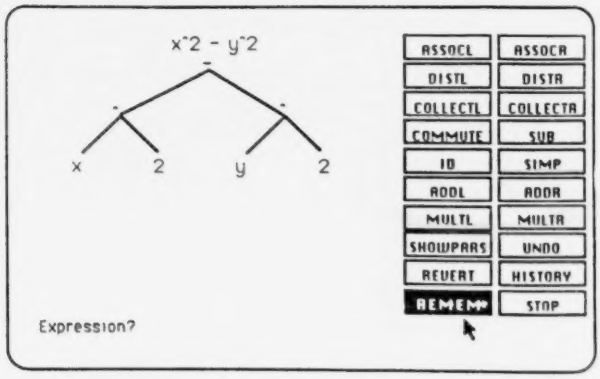


FIGURE 10.

Expressions has been used with eight average, leaving-seventh graders and with a class of preservice elementary school teachers. Students' use of **EXPRESSIONS** was guided by a workbook that contains graduated activities and exercises (see Thompson, 1985, 1987 for an explanation of graduated exercises in a Piagetian tradition). The exercises began with order of operations in arithmetic expressions, moved to derivations of equivalence of arithmetic expressions by use of field properties, and then moved to derivations of identities of algebraic expressions by use of field properties and identities.

Thompson (forthcoming) provides analyses of videotapes and computer-recorded interactions taken during the field trials with the seventh graders. Here I will note two observations. First, contrary to my expectation, students found expression trees to be quite intuitive. In fact, they soon preferred working with expression trees to working with expressions in sentential form (all exercises were presented in sentential format).

Second, when planning the pilot, I did not have a clear sense for how to motivate the use of letters in expressions. There was not time for an elaborated treatment, so I decided to just begin using letters and listen closely to the students reactions—and plan accordingly in succeeding studies. To my amazement, students were not bothered by the introduction of letters in expressions. They quickly saw letters as placeholders for substructures in a tree. For example, six of the eight students explained that in applying the associative property to (3+5)+(7+9) to get 3+(5 + (7+9)), the subexpression (7+9) "is just like c" in (a + b) + c.

The fact that students could see that two expressions like (a+b) + c and (3+5) + (7+9) were both appropriate for the application of the ASSOCL transformation suggests that they were developing a generalized concept of variable—letters could stand for numbers, but in general they were placeholders within a structure and anything could be put in their place, including other expressions. Two students, when considering what identities to apply to (a - b)/c in order to obtain a/c – b/c, said that "(a - b) is like u" when comparing their expression to the identity "u/v = u * 1/v."

Frequently, students are bothered by the introduction of letters in expressions. Why? In their experience, an expression is there to be evaluated, and they cannot evaluate an expression having a letter in it. The seventh graders who used Expressions had a different kind of experience with expressions. In their use of Expressions, an expression was there to be manipulated. The presence of a letter in an expression did not affect their ability to manipulate it.

My hypothesis about why students used substitution spontaneously when comparing a complex expression with one that had the same structure but less complexity is this: These students realized the general purpose of letters because of the nature of their activities with arithmetic expressions. When transforming an arithmetic expression to obtain some goal expression, the particular numbers did not matter—only the current and goal expressions' structures constrained their choices. Also, the arithmetic exercises included many instances of transformations that left subexpressions intact. In working these exercises, students became used to ignoring, for the moment, the details of a subexpression and treating it as a single item in the super-expression. They were thinking in terms of substituting an expression for a variable. A cognitive foundation had been laid for explicit substitution of expressions for letters.

The **EXPRESSIONS** program will be used in Fall of 1987 with an eighth-grade pre-algebra class. In that study we will examine several questions, principally dealing with how students conceptualize the subject matter as a result of their interactions with **EXPRESSIONS**. Another set of questions deal with students' facility with paper-and-pencil work after they have used this program. It could very well be that skills developed in the context of **EXPRESSIONS** are bound to that context, in that the program does not allow students to do anything illegal. There is the distinct possibility that, left to their own devices, students will create many of the "off-the-wall" misconceptions that have been widely reported in the research literature.

From the little work that has been done so far, it appears that "direct engagement" software for teaching and learning mathematics has a rich potential for improving the ways students conceptualize and understand mathematics. However, there is much to be learned and many refinements to make in the approach before widespread introduction in the schools can be expected.

REFERENCES

Draper, S. W., & Norman, D. A. (1986). Introduction. In D. A. Norman, & S. W. Draper (Eds.), *User centered system design*. Hillsdale, NJ: Erlbaum.

Thompson, P. (1985). Experience, learning, and problem solving: Considerations in developing mathematics curricula. In E. A. Silver (Ed.), *Learning and teaching mathematical problem solving: Multiple research perspectives*. Hillsdale, NJ: Erlbaum.

Thompson, P. (1987). Mathematical microworlds and intelligent computer-assisted instruction. In G. Kearsley (Ed.), *Artificial intelligence and education*. Reading, MA: Addison-Wesley.

Thompson, P. (forthcoming). The effect of multiple, linked representational systems on students' understanding and skill.

WORKING TOWARD COMPUTERIZED PHYSICAL EDUCATION

By: Warren Toneman, Hinsdale Elementary SD #181, Hinsdale, IL

Physical educators through intramurals, coaching, and regular classwork are very number oriented. Batting and fielding averages, shooting percentages, fitness percentiles, won/loss records, squad rotations, leading scorers and rebounders, etc. are just some of the ways numbers enter our curriculum. More and more physical education teachers are finding that computers can help store and report this data to us.

Although many teachers have tracked physical fitness records for each student, little has been done in other areas of our curriculum. This has started to change because of the educational reform legislation of August, 1985. Public Act 84-126, effective August 1, 1985, amended The School Code of Illinois to include for the first time in the state's history, a definition of schooling and a requirement that the goals for learning be identified and assessed. The legislation defines the primary purpose of schooling as the transmission of knowledge and culture through which children learn in the following areas: language arts, mathematics, biological and physical sciences, social sciences, fine arts, and physical development and health. Further, it states that school districts must assess student learning at least at grades 3, 6, 8, and 10.

In this reform act daily physical education is no longer mandated. However, physical education is at least one of the seven areas that school districts must develop goals and objectives for. It is now up to the Physical Educator to take a lead in developing these goals and assessments.

PHYSICAL FITNESS

At Oak School, I began to tabulate my student fitness scores three years ago. I was very interested in two areas. The first area concerned general fitness of each class. By using Appleworks spreadsheet function, I was able to enter data and compile report forms that showed how each class performed the six items of the AAHPER Fitness Test. I was able to obtain class averages, grade averages, male and female averages, and finally the school average for each year. I am trying to now prepare norms for my school. I feel they would be more important for my students and I than national norms.

There are programs available that allow you to enter each students raw scores and then get a print out with rankings and percentiles. The problem is that these programs won't allow me to group by class, grade, or sex. There is also no means to store students scores from year to year for comparison. If a program could be developed to score, group, and analyze, it would be a tremendous asset for physical educators.

The second area I am trying to develop is a tracking system for my students from grade 3 to 6. In the Spring of each year, I send the enclosed graph for each individual student home for parent review. I would like to use a similar graph and have the program print out the information. I think a bar graph would be appropriate with a column for each test and year. This would be a time saving device as well as an excellent visual aid.

Following is a sample of my Appleworks spreadsheet report (Needer Grade 6) and individual test graph (Adrienne Biggert) for clarification.

COMPETENCY TESTING

Although the state requires in P.A. 84-126 for each district to establish goals and objectives, Hinsdale District #181 already has a year round test of approximately 65 items in various sports. This Spring, we are going to take the Student Learner Objectives from the ISBE and implement the grade six outcomes in their appropriate areas. This will probably bring our test items to approximately one hundred. My next project will be to develop a program that will store all this data from fifth grade so I can better plan my sixth grade classes. I will want to know which students could dribble a soccer ball, serve a volleyball, etc. If I knew this information ahead of time, I could plan better lessons and do a much better job of individualizing my program. This is going to be a tremendous project but will be well worth the effort. The computer is the answer to the storage of data and planning. I can store a great deal of data, but I can't get the program to fill my needs or answer my questions:

Who can do this/who can't?

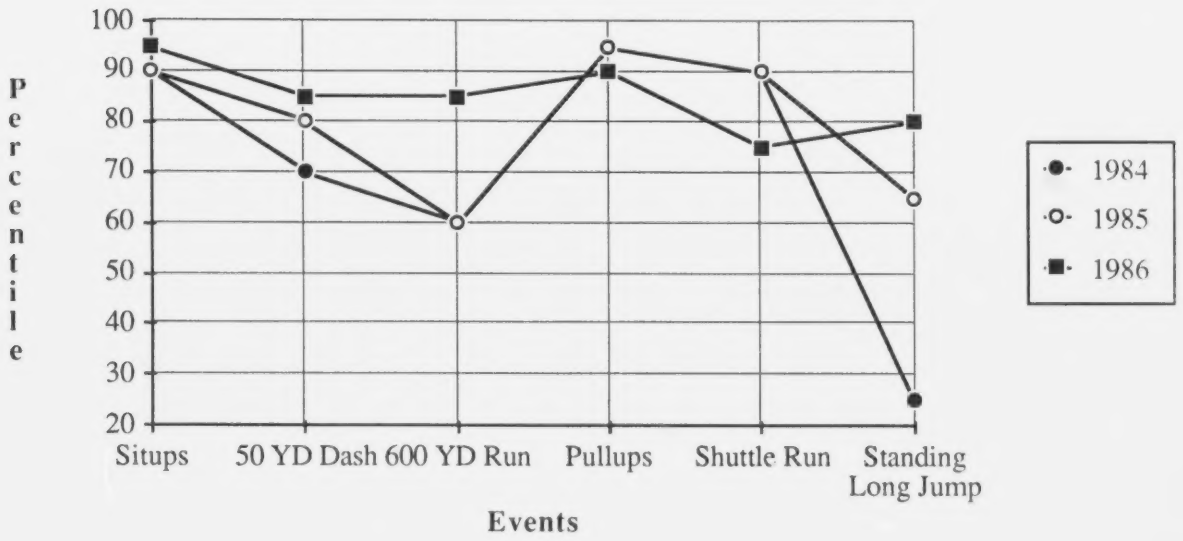
What % accomplished each item?

Who was absent and needs to make-up?

NEEDER GRADE 6

NAME	S	A	G	SIT	%	50	%	600	%	P/H	%	S.R.	%	JUMP	%
S. Ernest	F	11	N	42	90	8.0	55	176	50	16.7	70	10.3	90	72	95
K. Love	F	11	N	46	95	8.7	35	147	80	16.4	65	11.2	55	56	40
I. Martonffy	F	11	N	47	95	8.9	30	174	50	6.1	35	11.4	50	54	30
C. Reynolds	F	11	N	35	75	9.0	25	174	50	9.6	55	12.0	30	52	25
C. Larrick	F	11	N	45	95	7.3	95	130	95	12.6	60	10.0	95	64	75
A. Biggert	F	11	N	48	95	7.4	85	138	85	28.0	90	10.9	70	66	85
C. Curtis	F	11	N	38	85	8.0	60	174	40	5.6	40	11.1	60	50	15
K. Palmros	F	11	N	24	25	8.1	50	184	30	1.7	20	11.3	55	52	25
J. Reed	F	11	N	45	95	7.8	70	163	55	12.5	65	11.0	65	62	65
R. Robertson	F	11	N	40	85	7.9	65	170	45	10.0	55	11.4	50	59	50
E. Scheid	F	11	N	37	80	8.7	25	173	50	9.0	55	11.1	60	62	65
P. Sfikas	F	11	N	44	95	8.1	60	152	70	15.0	70	10.7	75	61	60
K. Shean	F	11	N	37	80	INJ		INJ		INJ				INJ	
J. Burnstein	M	11	N	34	50	9.4	5	290	5	0.0	25	13.2	5	54	20
J. Eggum	M	11	N	47	95	8.4	25	157	75	4.0	75	10.3	75	56	30
B. Hartney	M	11	N	43	85	7.8	65	158	40	0.0	25	11.1	40	57	30
D. Milonovich	M	11	N	38	65	8.1	50	169	55	0.0	25	10.8	55	56	30
B. Spirison	M	11	N	43	85	7.9	65	170	60	0.0	25	10.6	60	59	35
J. Beery	M	12	N	37	55	9.6	5	207	5	0.0	30	10.5	60	63	35
R. Canning	M	12	N	50	95	7.2	80	126	75	3.0	65	9.8	90	68	65
R. Callopy	M	12	N	40	70	7.9	75	160	20	0.0	30	11.1	35	66	55
A. Linn	M	12	N	45	85	7.3	80	132	65	5.0	85	9.8	90	64	45
B. Reed	M	12	N	39	65	8.4	20	169	15	0.0	30	11.5	20	49	5
J.C. Thatcher	M	12	N	48	90	7.3	80	137	55	0.0	30	10.0	85	62	35
T. Zorc	M	12	N	54	95	7.7	75	168	15	10.0	95	9.6	95	68	65
CLASS AVG.				42	81	8.1	53	167	49	6.9	51	10.9	61	60	45
MALE				43	78	8.1	52	170	40	2	45	10.7	59	60	38
FEMALE				41	84	8.2	55	163	58	11.9	57	11.0	63	59	52
SIXTH GRADE				41	78	8.1	54	162	52	8	48	10.8	63	60	46
SCHOOL AVERAGE				37	70	8.2	57	168	47	8	49	11.1	61	57	44

Individual Physical Fitness Performance



CREATING SERVICE LEVELS IN A COMPUTING INFORMATION CENTER

By: Gary W. Tubb, University of South Florida, Tampa, FL

MISSION STATEMENT: ACADEMIC AND USER SERVICES

The mission of Academic and User Services is to provide computer training, technical support, and problem solving services to the University community. These services are provided by the Information Center, Workstation Support, and Technical Services.

The central mission of all of these groups is to bring a University Computing Services client to a level of self-sufficiency and currency in computing on University Computing Services hardware and software. Advice and consultation on application problems, clarification of computing client's requirements, hardware and software configurations, and technical problems are services that these groups address.

MISSION STATEMENT: INFORMATION CENTER

The mission of the Information Center, as an integral component of University Computing Services, is to ensure the availability and reliability of documentation, training, and consulting, such that it will lead to a stable computing environment and concurrently promote customer independence, knowledge, skill, and self-sufficiency.

KEY RESULTS AREAS, INDICATORS, AND OBJECTIVES

Key Results Areas	Indicators	Objectives
Customer Satisfaction	User Survey	Maintain user satisfaction with UCS at a minimum of 95%.
Software Utilization	System Captured Statistics	Growth measurement in execution of supported software as it can be related to customer base.
Customer Independence	Reduction in no. of problem calls	Track number of reported problems and compare to earlier measured data from equivalent timeframe.
Training	Attendance & User Evaluation	Maintain user satisfaction at a minimum of 95%. Evaluate need for/quality of a course when average attendance drops below 50% of capacity.
Documentation Production	Availability & Customer Usage	Upon recognition of a need for a locally produced document, have it available for customer use in no more than 7 working days.

-CONTINUED-

KEY RESULTS AREAS, INDICATORS, AND OBJECTIVES

Key Results Areas	Indicators	Objectives
Documentation Acquisition	Availability	Upon recognition/customer request for a vendor supplied document, have it available for customer use in no more than 21 working days.
Communications	Reduction in no. of problem calls	Track number of reported problems and compare to earlier measured data from equivalent timeframe.
Demonstrations (Vendors)	Customer survey and attendance records	Record attendance and maintain satisfaction with demonstrations to which they have been invited at a minimum of 85%.

SCOPE

Users: Admin (Financial, Office Auto); Instruction (Faculty, Students: 1) Engr, 2) Bus, 3) Soc Sci, 4) Comp Sci); Research (Faculty, Resource Anal, Grad Students); External (County Schools, Corporate, Comm. Colleges)

Services/					
Documentation Production & Acquisition	3 / 3	1 / 1	2 / 2	4 / 1	17
Training	1 / 1	2 / 3	3 / 5	4 / 5	24
Communication	3 / 4	1 / 5	2 / 4	4 / 3	26
Consulting	3 / 2	1 / 4	2 / 1	4 / 2	19
Vendor Relations	2 / 7	1 / 7	3 / 6	4 / 7	37
Scheduling (312)	1 / 6	3 / 6	4 / 7	2 / 6	35
Internal System Development & Maintenance	2 / 5	1 / 2	3 / 3	4 / 4	24
	43	38	47	54	

Users: Admin Financial Office Auto	Instruction Faculty Students 1) Engr 2) Bus 3) Soc Sci 4) Comp Sci	Research Faculty Resource Anal Grad Students	External County Schools Corporate Comm. Colleges	
Services/				
Communication Dial-Up 4 / 3	1 / 1	2 / 1	3 / 1	16
Communication Cable Attached 1 / 1	3 / 4	2 / 3	4 / 4	22
Communication FIRN 4 / 4	1 / 2	2 / 4	3 / 2	22
Communication BITNET 2 / 2	3 / 3	1 / 2	4 / 3	20
21	18	17	24	

SCOPE (DETAILED)

1) Documentation Production & Acquisition

A) TIPs
B) PRIMERs
C) CICSTOM (CICS Terminal Operator's Manual)
D) Answer/DB Users' Information
E) UCS Newsletter
F) Maintenance of existing documentation
G) Maintenance of EXEC's, CLISTS's, & other related datasets
H) Maintenance of Mailing Lists for information dissemination
I) New User's Guide
J) Research and Testing
K) SLSS and other manuals
L) Publicizing the availability of documentation
M) Distribution
    1. Electronic
    2. Mail (Campus & U.S.)
    3. Manual Delivery

2) Training

A) Host Software
B) PC Software
C) Determine courses to be offered
D) Evaluation of training materials
E) Preparation of training materials
F) Preparation of necessary accompanying documentation
G) Scheduling
    1. Dates, Times, and Facilities
    2. Registration
    3. Certification
H) Training Evaluation strategies
I) Vendor Communications
    1. Keeping abreast of new training product offerings
    2. Phone calls/sales representatives, etc.
J) Facilitation of the purchase of training product offerings
K) Product returns
L) Distribution of training informations and materials

3) Consulting

A) Problem determination
B) Assimilate resources for resolutions
C) Suggest appropriate documentation
D) Encourage autonomy and self-sufficiency
E) Phone support
F) Testing user code
G) Mainframes
    1. SAS Institute
        a. Base SAS
        b. SAS/FSP
        c. SAS/GRAPH
        d. SAS/OR
    2. SPSS
    3. SPSSX
    4. Waterloo SCRIPT
    5. WYLBUR
        a. Editor
        b. Command Processor
        c. Exec Language
        d. USF EXECs
    6. TSO
        a. Editor
        b. Command Processor
        c. CLIST Language
        d. USF CLIST EXECs
    7. VM/CMS
        a. XEDIT
        b. XEDIT Macros
        c. REXX Language
        d. USF REXX EXECs
        e. CMS & CP
        f. Programming Languages

8. Student Languages
    a. BASIC
    b. COBOL
    c. Pascal
    d. FORTRAN 77
    e. WATFIV
    f. PL/I
9. OS/370 JCL
10. Utility Programs
    a. IEBGENER
    b. IEBCOPY
    c. WUNPRESS
    d. Tape Scan
    e. Bypass Label Processing
11. Tape Library Management System

H) Personal Computing (IBM PC's Compatibles)

1. PC/DOS (MS/DOS) -- Personal Computer Disk Operating System
2. DisplayWrite3 -- IBM Word Processing Software System
3. Lotus 1-2-3 -- Personal Computer Spreadsheet Package
4. dBASE III -- Personal Computer Database Package
5. 3270 Emulation
    a. IBM 3278/79 Emulation (Coax Attached)
    b. 5253 Attachment Program (Twinax Attached)
    c. SNA3270 Emulation (2400 BAUD/SDLC Dial-up)
6. Dial-up Access
    a. IBM 3101 Emulation (300/1200 BAUD)
    b. PC-TALK4 (300/1200 BAUD)
    c. RENEX Protocol Convertor (Full-Screen)
    d. PCI Protocol Convertor (Line-at-a-time)
7. OA/WP (IBM 5520) -- Document Interconversion
    a. 5520 Documents
    b. DisplayWrite 3 Documents
    c. PC Text Files
8. File Transfer
    a. IBM PC to MVS/TSO or VM/CMS
        1. 3278/79 Emulation
        2. 5253 Attachment
        3. SNA3270 Emulation
    b. IBM PC to MVS/WYLBUR
        1. PC/Talk IV (Line-at-a-time)

4) Communications
    A) Dial-up
        1. PCI (Line-at-a-time)
        2. RENEX (Full-Screen)
        3. SNA3270 (SDLC--PC Network Emulation)
    B) Cable Attached
        1. 3278/79 Emulation
        2. 5253 Emulation

C) Network
    1. FIRN
    2. BITNET
D) Distribution/maintenance of PC-TALK4 diskettes and documentation
E) Testing communications packages, etc.

5) Vendor Relations
    A) Vendor Demonstrations/Presentations
        1. Organization
        2. Scheduling
        3. Overseeing/Follow through
    B) Communications
        1. Products/Services updates
        2. Maintenance of target user mailing lists
        3. Correspondence (to Vendors and Users)
        4. Discussions with others about ideas for new demonstrations
    C) Distribution of information about demonstrations to users
    D) Insure availability of equipment and supplies

6) Office Procedure (Scheduling--312)
    A) Scheduling of SVC 312
        1. Room preparation (equipment, etc.)
        2. Internal UCS groups and UNIFTRAN (Meetings)
        3. USF Departments
            a. Training for SAMAS, DISOSS, etc.
        4. In-house training
            a. Lotus 1-2-3, Wylbur, etc.
        5. Vendor Demonstrations/Presentations
        6. External Groups
            a. Auditor General
            b. Weekend College, Saturday's
            c. IBM PC User's Group
        7. Communications with all involved parties
    B) PC Portable check-out forms and procedures
    C) Project documentation
    D) PC Club Public Domain software library handling
    E) Other required reports, etc.

7) Internal System Development & Maintenance
    A) EXEC's
        1. VM/CMS -- REXX
        2. Wylbur
    B) CLIST's
        1. TSO
    C) Student Consultant Handbook
    D) Presentation/tracking systems
        1. TIPs
        2. Primers
        3. UCS Newsletter
        4. CICS Terminal Operator's Manual
        5. Answer/DB Authorized User List

ROBOTICS AND LOGO STRUCTURES

By: Gary W. Tubb, University of South Florida, Tampa, FL

The essence of this paper is that robotics may be used as a motivator for fundamental science, computer education, and mathematics in upper elementary and and middle school. In the first section of the paper, fundamental questions relevant to robotics are answered. The second section deals with concrete examples of robotics that may be easily used in the classroom to teach science, computing, and mathematics concepts.

INTRODUCTION

WHAT IS A ROBOT?

Joseph Engelberger, the father of industrial robotics (Unimation), believes that "a robot is simply a machine that does things that people would otherwise do." Most people in industry view robotics as an extension of automation and an additional tool for man to use in his quest for quality and productivity in manufacturing. The most often used definition of robotics follows:

> A robot is a reprogrammable multifunctional manipulator designed to move materials, parts, tools, or specialized devices, through variable programmed motions for the performance of a variety of tasks. (The Robot Institute of America)

Androids are robots that closely resemble human beings in action, shape, and capability. Human simulators for medical training purposes are examples of androids.

WHY ARE ROBOTS USED?

Robots can be built to do more precise work than humans, as well as more rugged work. Robots can do dangerous work and go where humans cannot go. Once a task can be programmed to be done by a robot, the task can be completed 24 hours a day with great accuracy day after day.

WHAT CAN ROBOTS DO?

Robots can lift and carry heavy objects. Poisonous gases and fumes do not bother robots. But, robots are used most often for dangerous jobs like dismantling bombs, patrolling prisons, putting out fires, mining the sea floor, or collecting scientific data from Mars.

HOW DO ROBOTS AFFECT PEOPLE?

Studies have shown that one robot can replace from 2 to 5 human workers on an assembly line. The average automobile assembly worker's pay is approximately $16.00 per hour. In comparison, the operating costs of robots are between $4.00 and $5.00 an hour for a six-axis, servo-controlled, computer-driven (sophisticated) robot that is amortized over 8 years (Blaesi, 1984). In 1970, 200 industrial robots were in operation in the United States. By 1983, the number had increased to approximately 7,000, less than half of the industrial robots in use in the world. By 1990, over 90,000 industrial robots are projected to be in use. In most cases, the robot will replace humans in those jobs which are normally hazardous or demeaning.

CONCRETE EXAMPLES OF ROBOTICS

MINIMUM MATERIALS

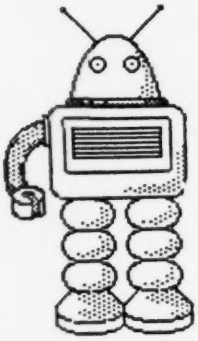
All to often we teach science and mathematics without concretely demonstrating the context from which fundamental concepts emerge. Robotics can be used as the context, the motivator, from which these concepts may be drawn. Two activities that require a minimum of materials, yet emphasize the scientific method, are presented in the appendix. The purpose of the activities is to discriminate among the tasks that robots can readily do, and provide a procedure that tests the quality of discrimination.

The first activity asks what a one-arm robot can do. The second activity asks what a two-arm robot can do. The procedure for testing the tasks that a two-arm robot can do, requires that two children sit side-by-side with one arm over the shoulder of their friend. One child must use his left hand with the other child's right hand to accomplish the task. This process imitates the scientific process and characterizes the communication difficulties of multiple feedback and commands. The purpose - investigate what the two-arm robot can do? The hypothesis - can the two-arm robot tie his tennis shoe laces? The procedure - using clothes pins as grippers, the two children attempt to tie shoe laces. Finally, the children come to a conclusion.

The process of testing whether a robot can accomplish a task is further investigated under the limitations of each of the human senses. The children try the same task with gloves on (sense of touch), blindfolded, and without talking. Investigating each of the senses and how you create a robot with those senses requires basic science and mathematics. A simple switch made of aluminum foil between sponges can be used to simulate touch (stopping the closing of grippers at some pressure). A cadmium sulfide cell will vary in resistance when exposed to various levels of light, and thus may be used as a crude robotic vision system that tracks stars or senses a forest fire. The very structure of a robot's arm introduces the idea of levers.

COMPUTER INTERFACED ROBOTICS

In the appendix you will find a list of robot toys and interfaces that may be used in the classroom. A kit that may meet the needs for those already with some experience is the Fischertechnik Robotic Computing Kit. Two different robotic arms may be constructed from the kit: a coordinate robot arm, and a teachable robot (pendant control). Other interesting process controlled devices are: a traffic light with interrupt for simulating a pedestrian changing the timing pattern upon demand; a materials lift (or elevator) that demonstrates queue concepts and priority requests, an aerial rotor and solar cell tracking device that demonstrate the mechanics of dampening motor oscillations and feedback.



MRS. MITCHELL'S
CLASS ROBOT

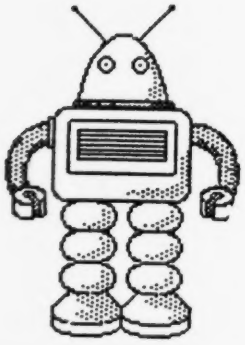
ONE-ARM ROBOT CAN'T TIE SHOE LACES

TAMPA, FL (ME) - In an effort to give robots a human touch, Mrs. A. Mitchell's fourth grade class at Morganwood Elementary put tennis shoes on their class robot. The one-arm robot could not tie his shoe laces.

Decide if a one-arm robot can do these things.

1. Pick up cans?
2. Brush my hair?
3. Sort socks?
4. Balance a penny on edge?
5. Set the table?
6. Button my shirt?

List six things that a one-arm robot can do.



RETROFIT ROBOT

ELEMENTARY STUDENTS CAN'T TIE SHOE LACES USING TWO HANDS!!!

TAMPA, FL (ME) - Teaching a two-arm robot to tie shoe laces is difficult. A fourth grade class at Morganwood Elementary tried to tie their shoe laces like they would teach a robot. Two children sat side-by-side with one arm on the shoulder of their friend. With the right hand of one child and the left hand of the other, they tried to tie their shoe laces. There is concern whether these children should be promoted to the fifth grade - since they could not tie their shoe laces.

Decide if it would be easier for the children if:

1. They could not talk?
2. They wore blindfolds?
3. They wore gloves?
4. They could only use two fingers (like pinchers)?

Could a one-arm robot take apart and put back together a ballpoint pen? List things that a two-arm robot can do (test your idea with a friend like you did when you tied your shoe laces).

SOME SUPPLIERS OF ROBOTS AND THEIR PRODUCTS

(Prices quoted are approximate and are given as a guideline.)

TOYS

Computer Games Ltd (CGL), CGL House, Goldings Hill, Loughton, Essex.
George the Compurobot. (programmable through a membrane keypad on top) £25.

Commotion, 241 Green Street, Enfield, Middlesex EN3 7SJ.
Beasty Truck (controlled by BBC Micro, caterpillar-tracked vehicle) £120
Movits
Wizard (controlled by Commodore 64) £75
Memocon Crawler (controlled through pendant) £35
Navius (programmed by colored disc) £20
Medusa (four legs, sound sensor) £20
Piper-mouse (three wheels, whistle activated) £20
Monkey (sound activated, climbs wire) £10
Line Tracer (infra-red sensor) £18
Peppy (bump sensor) £17
Avoider (infra-red sensor) £30
Circular (remote-controlled) £30

Fischertechnik, imported by Economatics (see below)
Kit system, needs interface and micro for control, from £65

Milton Bradley Ltd., Spencer House, 23 Sheen Road, Richmond, Surrey.
Robotix (three kits, with own control system) £20 to £80

Tomy Toys Ltd., Wells House, 231 High Street, Sutton, Surrey.
Omnobot (radio-controlled mobile with two simple arms) £190
Verbot (voice-controlled mobile, see text) £50
Dingbot (bump-sensitive mobile) £10

TURTLES

Clwyd Technics Ltd., Unit 4B, Antelope Industrial Estate, Rhydymwyn, Clwyd CH7 5JH.
Trekker (comprehensive software and documentation) £125

Economatics Ltd., 4 Orgreave Crescent, Dore House Industrial Estate, Handworth, Sheffield S13 9NQ
BBC Buggy (specifically for the BBC-B, light and bump sensors, grabber) £200

IGR, Unit 208, Highbury Workshop, 22 Highbury Grove London N5
Zero 2 (Update of the Zeaker turtle; runs under Logo on many popular micros) Kit £80

Jessop-Ralph Ltd., Unit 5, 7 Long Street, London E2
Jessop Turtle (traditional design) £200

Penman Products Ltd., 8 Hazlewood Close, Worthing, W. Sussex BN14 8NP
Penman Plotter (combines functions of a turtle, a plotter and a mouse for input, three pens) £230

ROBOT ARMS

Powertran Cybernetics Ltd., Portway Industrial Estate, Andover, Hants SP10 3NN
Hebot II (traditional design, controlled by ZX81, amongst others) £100
Valiant Designs, Ltd., Unit 14, Park House, 140 Battersea Park Rd., London SW11
Valiant Turtle (infra-red control, runs under Logo, looks like a real turtle!) £150

KITS

Colne Robotics Ltd., Beufort Rd, E. Twickenham, Mddx TW1 2PQ
Armdroid (5 axes, stepper motors; one of the earliest designs) £500
Commotion as above
Beasty (3 axes, servo motors; expandable) £115
Cyber Robototics Ltd., Stone, Staffs ST15 0SA
Cyber 310 (5 axes, stepper motors; special software suite) £650
Cybernetic Applications Ltd., West Portway Ind. Estate, Andover Hants SP10 3LF
Mentor (5 axes, servos; controlled by miniature model of the arm) £500
Neptune 1 (5 axes, water hydraulic) £1600 kit, £2500 built
Neptune 2 (6 axes; touch sensor control) £2200 kit, £3200 built
Serpent (5 axes, SCARA, Selective Compliane Assembly Robot Arm; moves in a horizontal rather than vertical plane) £2000
Feedback Instruments Ltd., Park Road, Crowborough E. Sussex TN6 2QR
HRA 933 (5 axes, oil hydraulic; ready-built version of Genesis P101) £2000
HRA 934 (5 axes, oil hydrauiic; built Genesis P102) £2700
Pedro (4 axes, servos and steppers; educational workcell SCARA design)
L.J. Electronics Ltd., Francis Way, Bowthorpe Ind. Estate, Norwich NR5 9JA
Atlas (5 axes, steppers; powerful teaching aid) £1750
Mitsubishi, Hertford Place, Maple Cross, Herts WD3 2BJ
Movemaster (5 axes, servos; intended for light industrial use) £6000
Powertran Cybernectics Ltd., as above
Genesis P101 (5 axes, oil hydraulic; first to use hydraulics) Kit, £1000
Genesis P102 (more memory than P101) Kit, £1350
Micrograsp (4 axes, servos; one of the earliest low-cost designs) £300
IVAX (4 axes, servos; SCARA design) "under £1000"
Reekie Research Ltd., Beaufort Road, Twickenham, Mddx TW1 2PQ
Cepek 1 (4 axes, servos; SCARA design) £4000
Reekie Arm (5 axes, steppers; similar to Armdroid) £250
Remcon Electronics Ltd., P.O. Box 81, Chislehurst, Kent BR7 6LP
Teach Robot (5 axes, servos; uses screw transmission) £400
Robot City Technology Ltd., 35 Clarke Road, Mount Farm Ins. Est. Milton Keynes
Alfred (5 axes, servos; many add-ons promised) £200
Robotec Ltd., & Hartley Court, Norton Street, Radford, Nottingham NG7 3AN
Hero 1 (company specializes in industrial robots)
Syke Instrumentation Co. Ltd., 117 Station Road, Liss, Hants
Teachmover (5 axes, steppers; early U.S. design) £2000
Minimover (5 axes, steppers; smaller version of Teachmover)

Systems Control Ltd., Perry Avenue, Teeside Ind. Est. Thornaby, Cleveland TS17 9LN

Smart Arm (5 axes, servos; early British design) £1500

TecQuipment International Ltd., Bonsall Street, Long Eaton, Nottingham NG10 2AN

MA 2000 (6 axes, servos; designed for Open University courses) £3600

Universal Machine Intelligence Ltd., Royal Victoria Patriotic Building, Trinity Road, London SW18 3SX

RTX (6 axes, servos; modified SCARA design with software for control from IBM PC)

LOGO AND MIDDLE SCHOOL MATHEMATICS

By: Sandra V. Turner and Michael L. Land, National College, Evanston, IL

PURPOSE

The purpose of this study was twofold:

1) To investigate the effect of learning Logo on middle-school students' understanding of specific mathematical concepts: properties of polygons, angle measurement, estimation, rectangular coordinate system, negative numbers, and variables; and

2) To investigate the effect of learning Logo on middle-school students' level of cognitive development from a Piagetian perspective (concrete operations, transitional, formal operations).

BACKGROUND

Numerous individuals have made claims regarding the cognitive benefits of learning Logo. The published research on the effects of Logo, however, is conflicting. Positive results (Table 1) have been reported by Gorman & Bourne (1984) while Gaffney (1983), Horner (1984), Milojkovic (1984), Pea & Kurland (1983), Rood (1983), Seidman (1980), and Shaw (1984) reported no significant effects due to Logo. In addition, Clements (1985), Odom (1984), Reimer (1985), and van Dyke (1984) reported both significant effects or correlations.

One of the reasons for conflicting evidence on the cognitive benefits of learning Logo may be weaknesses in the design of some of the studies. Some studies had no control group or a small number of subjects; many had short treatment periods. Another possible explanation for the conflicting evidence is that the majority of studies have assumed that the relationship between Logo and learning, if it exists, is a straight-line cause-and-effect relationship. If there were a significant straight-line, cause-effect relationship between Logo and learning, then any of the three types of research designs (correlation, experimental, and criterion group) should reveal that relationship. However, if there were a curvilinear, cause-effect relationship, then a researcher might not find evidence for the relationship using a traditional experimental design. The majority of the reported research studies on Logo use an experimental or quasi-experimental design, a design which yields significant findings for a straight-line cause-effect relationship, but not necessarily for a curvilinear cause-effect relationship (Table 2). Therefore, one possible explanation for the highly inconsistent findings is that the relationship between Logo and learning is not a simple straight-line relationship.

POTENTIAL TYPE OF SIGNIFICANT RELATIONSHIP BETWEEN LOGO AND LEARNING	TYPE OF DESIGN (finding expected)		
	Correlation	Experimental	Criterion Group
A) No relationship	NS	NS	NS
B) Relationship			
(1) Cause-effect			
[a] straight-line	Sig	Sig	Sig
[b] Curvilinear	NS or Sig	NS	Sig
C) Common causes			
(1) straight-line	Sig	NS	Sig
(2) Curvilinear	NS	NS	Sig

TABLE 2

METHODOLOGY

Middle school students (n = 91) in four intact classes in two Chicago public schools learned Logo for one hour a week for 16 weeks in place of their regular mathematics curriculum. A control group of students (n = 90) in different Chicago schools did not participate in the Logo program, but received the full allotted time for their regular mathematics curriculum. Students in both the experimental (Logo) and control groups were pretested and posttested to determine: a) their achievement of selected mathematics concepts, and b) their level of cognitive development. In addition, the students in the treatment group took a posttest to measure their understanding of Logo concepts.

CURRICULUM

The goal of the curriculum was to use Logo to supplement the learning of these specific mathematics concepts: estimation of angles and distance, properties of polygons, variables, rectangular coordinate system, negative numbers, perimeter, and symmetry. These topics were part of the middle school mathematics curriculum in each school, but the classroom teachers had planned to cover them only briefly. At the same time, students learned Logo programming concepts: iteration (use of REPEAT), defining procedures, use of subprocedures, variables, planning, modifying, and debugging.

Classes were held in the school's computer lab with two students sharing a computer and working jointly on most activities. The two researchers, both experienced Logo teachers, shared the planning and teaching responsibilities. Each one-hour lesson included time for both structured learning activities and exploratory, open-ended activities.

Students (n = 30) in one of the four experimental classes had participated in a similar Logo mathematics program the previous year (one hour per week for ten weeks), and a more advanced curriculum was planned for them. Since they were studying pre-algebra, they used Logo to learn about order of operations and evaluating algebraic expressions as well as more advanced work with variables. Their Logo environment included elementary list processing in addition to turtle graphics.

ASSESSMENT INSTRUMENTS

The following instruments were used to measure the dependent variables:

Cognitive development. The Social Sciences Piagetian Inventory (SSPI) (Carter & Ormrod, 1982) is a 30-item, paper and pencil multiple choice test which measures cognitive development from a Piagetian perspective. The test is designed for persons age 10 or older, although younger children must be of average intelligence or higher. The Dale-Chall reading level of the test is grades 5-6.

Scores on the SSPI classify individuals into three broad Piagetian categories: concrete operations, formal operations, or in a transitional stage between these two.

Items for the SSPI were written to measure specific skills defined by Piaget for concrete and formal operations, such as reversibility, conservation of area, transitivity, proportional reasoning, and isolating and controlling variables. To provide evidence of the concurrent validity of the test, 30 students ages 10 through 13 were given both the SSPI and individual clinical interviews, and the classifications agreed 60% of the time. A test-retest reliability study with one-week delay resulted in a reliability coefficient of .87 for a sample of students in grades 5-9 (N = 141).

Mathematical concepts. The researchers developed a 22-item multiple choice test designed to measure students' understanding of several specific mathematical concepts assumed to be related to Logo. The concepts included in the test are: estimation of angles, estimation of distance and area, properties of polygons, rectangular coordinate system, variables, negative numbers, symmetry, perimeter, and recognition of number patterns. The test was piloted the previous year with 110 students in grades 5-8, and several items were revised on the basis of the pilot data. The test-retest reliability coefficient for the test is .75 for a sample of 181 students in grades 5-8 (six-month delay).

Logo concepts. In order to assess students' conceptual understanding of Logo, the researchers developed a 20-item multiple choice test. The items of the Logo test ask the student to predict the output of a given Logo instruction, to determine which instruction has the given output, or to fill in a missing instruction in order to produce a given output. The test, which is limited to turtle graphics, covers iteration (the use of REPEAT), the rule of 360, procedures, and variables, but it does not measure students' knowledge of Logo syntax or the operating system. The Spearman-Brown split-half reliability coefficient for the test is .77 for a sample of 22 students in grades 5-6.

RESULTS AND CONCLUSIONS

Q1: What is the effect of learning Logo on students' achievement in selected mathematics/geometry concepts?

A1: There were no significant differences between the control and experimental groups on the total score on the mathematics test or on two component subtests. However, this result may be educationally significant because there were no significant differences even though the Logo group spent 20 percent less time than the control group studying the regular mathematics curriculum. Students in the Logo group learned as much mathematics (as

measured by this particular test) as the control group, plus they had the additional benefits of working with Logo.

Q2: Is there a correlation between learning Logo and knowledge of selected mathematics concepts?

A2: There was a significant, positive correlation ($r = .58$, $p < .001$) between students' knowledge of mathematics and their knowledge of Logo. However, there was no significant correlation between the amount of mathematics they learned during the treatment and their knowledge of Logo. In other words, if they achieved well in one area, they tended to achieve well in the other area. Apparently, there is a common factor or factors affecting both achievement in mathematics and achievement in Logo.

Q3: What is the effect of learning Logo on students' level of cognitive development?

A3: Both the experimental and control groups had growth in cognitive development, but there were no significant differences between control and experimental groups on the SSPI. As a further observation, both groups had more students at the concrete level of cognitive development than one would expect based on chronological age and Piaget's theory.

Q4: What is the relationship between learning Logo and students' level of cognitive development?

A4: There was a significant, positive correlation ($r = .54$, $p < .001$) between Logo achievement and level of cognitive development. Thus, the students who learned more Logo tended to be at a higher level of cognitive development.

When we applied a criterion group design (High Logo versus Low Logo achievers), we found a significant difference ($F(1,30) = 6.21$, $p < .05$) in change in cognitive development in favor of the High Logo group. Students who learned the most Logo had significantly greater growth in cognitive development, compared to students who learned a minimal amount of Logo. This finding suggests that students need to reach a threshold in their understanding of Logo in order for it to be accompanied by growth in cognitive development.

Q5: Can teachers predict students' success in Logo?

A5: There was no significant correlation between the classroom teacher's estimate of a student's success in Logo and the student's actual Logo achievement. However, the teachers were able to predict which students would do very well (High Logo group) and which would do poorly (Low Logo group).

Q6: Were teachers biased on the basis of gender in their predictions of students' success in Logo?

A6: There were no gender differences in the classroom teachers' predictions of their students' success in Logo.

LIMITATIONS OF THIS STUDY

When generalizing the results of this study, the following limitations should be kept in mind:

1. The students were not randomly placed into the control and experimental groups and the control and experimental groups were not from the same school.

2. Students in one of the experimental schools were all above average in academic ability and those in the second experimental school were generally below grade level, thereby resulting in a bimodel experimental group.

3. Validity studies have not been done on the Logo and mathematics instruments, although both were piloted.

SUMMARY

The purpose of this study was to investigate the effect of learning Logo on middle-school students' knowledge of selected mathematics concepts and on their level of cognitive development. There were no significant differences between those who used Logo and those who did not, on their understanding the mathematics concepts, or on their growth in cognitive development.

When we compared those who learned the most Logo with those who learned only a minimal amount of Logo, there were significant differences on their growth in cognitive development.

These findings suggest that the relationship between Logo and learning mathematics and between learning Logo and cognitive development is not a straight-line cause-and-effect relationship, but may be curvilinear.

Logo, mathematics, and cognitive development apparently share a common factor. Growth in cognitive development appears to be closely associated with Logo achievement after a threshold has been reached in students' understanding of Logo.

REFERENCES

Carter, K. R. & Ormrod, J. E. (1982). Acquisition of formal operations by intellectually gifted children. **Gifted Child Quarterly, 26**, 110-115.

Clements, D. H. (1985). Logo programming: Can it change how children think? **Electronic Learning, 4,** 74-76.

Gaffney, C. R. (1984). Computer languages: Tools for problem-solving? The University of Texas at Arlington, **DAI, 45/09-B,** 3097.

Gorman, H., Jr. & Bourne, L. E., Jr. (1983). Learning to think by learning Logo: Rule learning in third-grade computer programmers. **Bulletin of the Psychonomonic Society, 21**, 165-167.

Milojkovic, J. D. (1984). Children learning computer programming: Cognitive and motivational consequences. Stanford University, **DAI, 45,** 385.

Odom, M. L. N. (1984). The effects of learning the computer programming language Logo on fifth and sixth grade students' skills of analysis, synthesis, and evaluation. University of Arkansas, **DAI, 45/08A**, 2390.

Pea, R. D. & Kurland, D. M. (1983). **Logo programming and the development of planning skills**. New York: Center for Children and Technology, Bank Street College, Tech. Rep. No. 16.

Reimer, G. (1985). The effects of a Logo computer of programming experience on readiness for first grade, creativity, and self concept: A pilot study in kindergarten. **AEDS Monitor, 23** (7,8), 8-12.

Rood, M. K. (1983). Locus of control, self-esteem, and the acquisition of problem-solving skills through the instruction of Logo and BASIC computer programming languages in gifted children. The University of Connecticut, **DAI, 45**, 469.

Seidman, R. H. (1980). The effects of learning the Logo computer programming language on logical reasoning in school children. Syracuse University, **DAI, 41**, 2249.

Shaw, D. G. (1984). The effects of learning to program a computer in BASIC or Logo on the problem-solving abilities of fifth grade students. North Texas State University, **DAI, 45/07A**, 1985.

Van Dyke, J. S. G. (1984). The relationship between Logo and creativity (brainstorming, imagery, problem-solving, creative process). Texas A & M University, **DAI, 45/09-A**, 2815.

Researcher	Design Used Corr	Exp	Crit Gp	Number of subjects	Factors Studied
Clements (1985)		X1		18	reflectivity [S] divergent thinking [S] metacognition [S] describing directions [S] vocabulary [NS] logical thinking (Piaget) [NS] verbal/number memory [NS]
Gaffney (1983)		X2		??	problem-solving [NS]
Gorman & Bourne (1984)		X		15	rule learning [S]
Milojkovic (1984)		X1, 2		58	cognitive tasks [NS]
Odom (1984)		X		36	Ross problem-solving: analysis [S] synthesis [NS] evaluation [S]
Pea & Kurland (1983)		X		32	planning [NS]
Reimer (1985)		X		16	visual discrimination visual motor skills [S] visual memory [S] number recognition [S] creativity [NS] self-concept [NS]
Rood (1983)		X2		76	problem-solving [NS] locus of control [NS] self-esteem [NS]
Seidman (1980)		X		42	conditional logic [NS]
Shaw (1984)		X3		132	reasoning skills [NS]

- CONTINUED -
TABLE 1

Researcher	Design Used Corr	Exp	Crit Gp	Number of Subjects	Factors Studied
van Dyke (1984)	X			27	verbal creativity & Logo achievement [S] figural creativity & Logo achievement [NS] intelligence & Logo achievement [S]
Turner & Land (1987)	X	X	X	181	math/geometry concepts [NS] math/geometry & Logo achievement [S] level of cognitive development [S] cognitive development & Logo achievement prediction & Logo achievement [S, NS]*

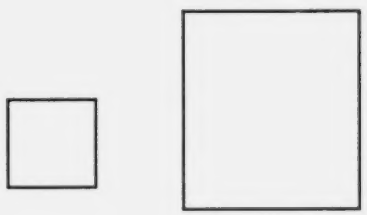
1 Logo vs CAI
2 Logo vs BASIC
3 Log, BASIC, none

* correlation not significant; criterion group significant

TABLE 1

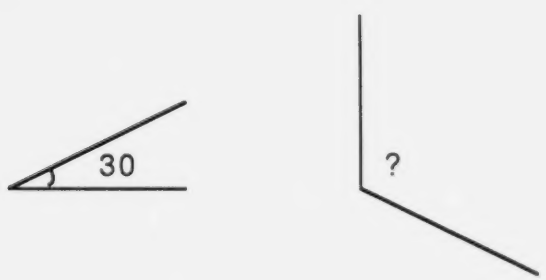
SELECTED ITEMS OF THE MATHEMATICS TEST

_____2. How many squares will fit inside the large square?



a) 4 b) 6 c) 8 d)

_____7. If the angle on the left has a measure of 30, estimate the measure of the angle on the right.



a) 60 b) 80 c) 100 d) 120

_____8. If the distance from A to B is 10 units, estimate the distance from B to C.

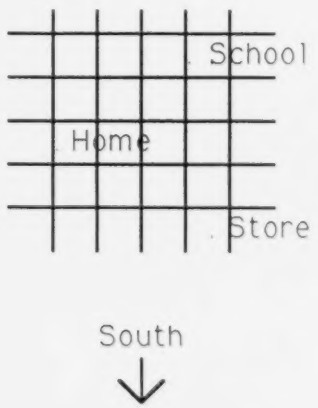
a) 35 b) 50 c) 75 d) 100

_____14. The same number belongs in both boxes. What is it?

$7 \times \square - 4 \times \square = 9$

a) 1 b) 2 c) 3 d) 4

_____20. You are at school facing south. Which set of directions is the shortest path to follow to get home?

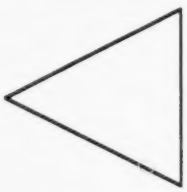


a) Forward 3 blocks, left 4 blocks
b) Forward 4 blocks, left 5 blocks
c) Forward 3 blocks, right 4 blocks
d) Forward 4 blocks, right 5 blocks

SAMPLE ITEMS FOR THE LOGO TEST

_____4. To draw a triangle, what number must be in the blank?

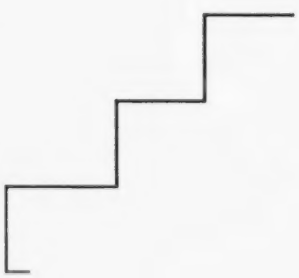
REPEAT 3 [FD 100 LT _____]



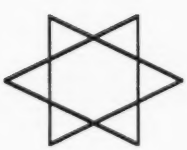
A. 30
B. 60
C. 90
D. 120

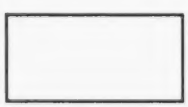
_____18. Which figure does this instruction draw?

REPEAT 3 [FD 30 RT 90 FD 30 LT 90]

A. 

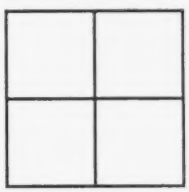
B.

C. 

D. 

_____19. In order to draw this window, what instruction must be in the blank?

```
TO WINDOW
REPEAT 4 [SQUARE 20 RT _______]
END
```

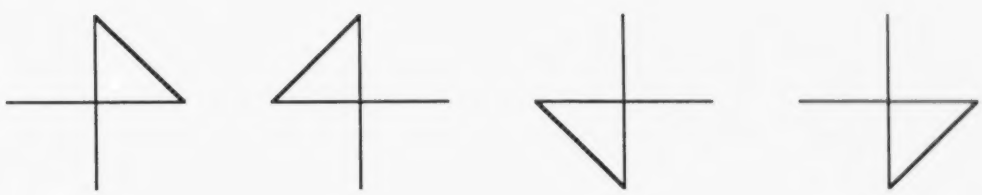


A. 20
B. 60
C. 90
D. 180

_____20. What figure does this procedure draw?

```
TO FIGURE
BACK 40
LEFT 45
FD 25
LEFT 45
BACK 40
END
```

A. B. C. D.



DESIGN FOR DECISION MAKING

By: Barbara T. Vincent, The Meadowbrook School, Weston, MA

PURPOSE

A colleague and I decided to design a curriculum which would concentrate on the goal of developing in sixth grade students an understanding of the complexities involved in decision making. We were specifically concerned with the "black and white" attitudes our students expressed regarding political issues, and with their apparently faulty model of the decision making process. We had observed that in discussions revolving around current events, these students believed that people in power made arbitrary and quick decisions. These same students judged such decisions in the same manner.

It has been suggested that students will persevere in maintaining faulty models, thus making instruction more difficult, and that instructional design must make explicit attempts to remediate students' misconceptions (Chen, 1985). Cognitive research suggests that learning can be significantly improved if instruction is designed so that the student can build on existing knowledge, develop new learning strategies, and correct learning problems (Jones, 1986, Wolfson, 1986). Our own teaching experience made us aware of the motivation that our students exhibit when working with computer software. This experience with interactive technology suggested that exposure to interactive media motivates students to acquire cognitive skills and knowledge at an earlier age (Chen, 1985). With the above in mind, we focused our design on explicit instruction, cognitive goals and the inclusion of interactive media.

The group involved consisted of 12 male and 9 female sixth grade students. All had extensive previous experience with the use of computers, including word processing, LOGO, CAI, and simulations. Some had experience with BASIC.

CHOICE OF SOFTWARE

Early in the year we had introduced and used **The Other Side**, a conflict resolution, computer simulation by Tom Snyder Productions. This simulation was used as a stand alone piece, and no attempt was made to integrate it with other parts of the curriculum. Our goal was to introduce and reinforce understanding of conflict resolution and this was successful.* As a result of this experience, we met David Doctorman, a HGSE doctoral candidate who works with Tom Snyder. David showed us a new piece of software called, **The Discussion Generator**. We were very impressed with this product and decided to make it the interactive piece in our design.

The Discussion Generator was designed for use in middle and high school history and social science curricula. The scenario for the program involves three interdependent countries dealing with a potential crisis. A small country, Strato, run by a king is dealing with a rebel force and possible civil war. Another country, Toro, may be aiding the rebels and appears to be in competition with a third country, Libros. Libros faces the question of how to deal with the rebel vs. king situation with the king demanding aid, both military and financial.

* A paper reporting this experience was written in January, 1986, for Harvard's Course P-490.

Participants, game players, in the situation are members of the governing forces of Libros and have been elected to serve for one year. They must first prioritize their goals among four which include achieving election the next year, protecting their citizens working in the country fighting the rebels, effecting an election of a popular leader in Stratos, and keeping Strato as an ally.

Once these goals are prioritized and fed into the computer, the participants respond to randomly generated situations. The program branches depending on the goal set by the participants, and choices are screen displayed. Before the governing group selects an action, they are referred to "experts" who provide different points of view. This information, found in the Resource Guide is textual. This text is presented in the form of short, historical references. For example, people such as F. D. Roosevelt, J. F. Kennedy, W. Churchill, and groups such as the SDS are cited in historical terms. Their actions and opinions during times of crisis are presented. Each member of the group(s) playing the game presents his/her point of view, relying on the particular expert he/she has been assigned. Discussion results in a consensus regarding which choice of action is to be selected. Once an action is chosen and keyed into the computer, the program displays the consequences and provides new choices for the changed situation. One time through the game involves five chances to make decisions and results in the simulated passage of one year.

This particular piece of software is unique in that it lends itself to curriculum integration across a wide range of dates and topics in a social science program. While other software packages are designed to reinforce skills, for example CAI, or, as in the case of programs such as **The Other Side**, to stand alone or force the teacher to design curriculum around it. **The Discussion Generator** does neither. It is a tool that a teacher can take from the shelf and use to motivate, instruct and provoke thought and collaboration among students in the midst of a predesigned unit. It is designed to make maximum use of the teacher as mediator in a one computer classroom.

The Discussion Generator seemed to be a good vehicle for teaching to content, transfer and motivation. The content could be the historical data learned from participation and related activities. Creativity could be fostered through role playing, and emphasis on generating different solutions for each situation. Transfer, while possible to imagine, would be hard to evaluate. We hoped that clearly communicated goals involving transfer, and continual discussions relating the choices and consequences of actions to other situations would provide the vehicle. While we focused on teaching to motivation, content and transfer, we also discussed the ideas embodied in teaching to rationality and intelligence.

CURRICULUM DESIGN

A class discussion was held during which students were explicitly informed as to the design of the unit and the intended goals. Emphasis was placed on the concept of transfer, and their previous experience with **The Other Side** was recalled and discussed in the light of transfer of skills. Students were asked to report verbally throughout the unit on their evidences of new awareness and the gaining of new knowledge. Active engagement in ongoing evaluation of the worth of all parts of the unit was stressed. Also discussed were learning styles, and students were asked to self-monitor their own methods of learning.

A pre/posttest was designed (Appendix A) consisting of three world crisis situations, each followed by four possible actions. Students were instructed to choose two of the situations and select one of the four options for action. They then defended their choices in writing. The posttest consisted of the same three situations. Students were instructed to answer the one situation they had not used during the pretest. They were also asked to select one of those they had previously answered and either defend the original choice or change it and explain the reasons for change.

A list of vocabulary words taken from the Resource Guide of **The Discussion Generator** (Appendix B). Students were asked to conduct an oral history project by discussing the vocabulary terms with parents and/or other adults. They were encouraged to tape responses or take notes. One class period was spent in a discussion during which the information gathered from this exercise was exchanged. Teacher emphasis was on the reinforcement of the content and on comparisons of historical situations.

A list of geographical references was taken from the same Resource Guide (Appendix C). One class period focused on finding the places on a world map and discussing, in the light of interdependence, distance between places, means of access, economic conditions and place in history.

A long term assignment was given in which each student was asked to choose a five year period in history and produce a "TIME" magazine with information as it may have appeared during the chosen time frame (Appendix D). The goals of balanced, objective reporting were emphasized. Models of "yellow journalism", biased reporting and propaganda were introduced (Appendix E).

The Discussion Generator was introduced and students played it first in small groups of four, second as a class unit and then once again in the same small groups.

Over a two week period, daily class time was devoted to work on the "TIME" magazines. Research was conducted and two teachers worked with individual students consciously provoking students to make historical comparisons and to develop depth understanding of their research results. Emphasis was placed on the creative acts of drawing political cartoons and illustrating their news stories. Other teachers from different disciplines were asked to act as advisors for certain students who needed help with specific aspects of the assignment.

Fourteen videos, each twenty minutes in length, involving contemporary history were made available to students as research tools (Appendix F). They were free to view them in school or to take them home overnight.

A variety of print materials were made available in the classroom. Bulletin boards were devoted to time lines, historical data and comparisons. The School Librarian collected print materials and set them aside for use by these students.

A feature length film, "The Mouse That Roared", was shown and discussed in the light of interdependence of countries in the world.

Upon completion of the magazines, the students exchanged and shared their final products. The magazines were then distributed to other adults in the school and shown to the fifth grade class. Each student editor discussed the goals of the unit and his/her particular period piece with at least one adult. The final products were then displayed at the main entrance to the school.

Additional voluntary activities were suggested: the creation of a video school news broadcast produced for viewing by other students in the school, a debate involving the Libyan bombing presented to a school audience, the creation of a mini General Assembly of the United Nations discussing the nuclear accident in Russia and the creation of a slide tape demonstrating use of **The Discussion Generator**. Formats for the voluntary activities were explained, a teacher/advisor identified, and meeting times for initial discussion established.

The Discussion Generator was used again. Each student has the opportunity to play two or three rounds of the game.

SOFTWARE EVALUATION

As a result of a study that Malone (1980) conducted in which he interviewed sixty-five elementary school students about their computer game preferences, he designed a theory of intrinsically motivating instruction. The framework for this theory can be found in Table 1. (Malone, 1984).

TABLE 1. - FRAMEWORK FOR A THEORY OF INTRINSICALLY MOTIVATING INSTRUCTION

I. Challenge
- A. Goal
  1. Personally meaningful goals
  2. Obvious or easily generated goals
  3. Performance feedback
- B. Uncertain outcome
  1. Variable difficulty level
     a. Determined automatically
     b. Chosen by learner
     c. Determined by opponent's skill
  2. Multiple level goals
     a. Scorekeeping
     b. Speeded responses
  3. Hidden information
  4. Randomness
- C. Tools vs. toys
- D. Self-esteem

II. Fantasy
- A. Intrinsic and extrinsic fantasies
- B. Cognitive aspects of fantasies
- C. Emotional aspects of fantasies

III. Curiosity

Optimal level of informational complexity
- A. Sensory curiosity: audio and visual effects
- B. Cognitive curiosity
  1. "Good form" in knowledge structures
     a. Complete
     b. Consistent
     c. Parsimonious
  2. Informative feedback
     a. Surprising
     b. Constructive

Malone, Thomas W., "Toward a Theory of Intrinsically Motivating Instruction", \Instructional Software\, Walker and Hess, 1984, 85.

Malone's criteria provided us with a framework within which we could evaluate **The Discussion Generator** both by itself and in conjunction with the other parts of our curriculum.

In order for the game to be challenging, it should have goals and an uncertain outcome (Malone, 1984). In his study, Malone found that "the single feature of the computer games that correlated most strongly with preference was whether the game had a goal" (p. 85). The extrinsic goal of **The Discussion Generator** is to move through the decision making process in such a manner as to amass a score as close to 100 points as possible.

Players first prioritize goals from among four: (1) Get re-elected, (2) Keep Strato as an ally, (3) Protect Librotians on Strato, (4) Elect a popular leader in Strato. There are 40 possible points earned by achieving the first priority, 30 for the second, 20 for the third and 10 for the fourth. After the fifth and final move, the program reports what the situation is in the countries and a score based on achievement of the goals set by the players. The score is broken down so that players can see points earned for each goal. For example, the first line of the report might read, "Goal #1 Protect Librotians --- 10 points out of 40." Malone writes that a good goal should be personally meaningful. He cites a study by Morozova (1955) which emphasizes: (1) that it is important that the skills being taught are means to achieving the goal, not the end in and of themselves, and (2) the goal being part of a fantasy and being one with which the player can identify. He also cites Csikszentmihalyi's (1975) two types of goals: fixed goals and emergent goals.

The skills being taught through **The Discussion Generator**, an understanding of the complexities involved in decision making and the principles involved with problem solving, are not the goals as seen by the players, but rather are the means to achieve the goal. The goal is certainly part of a fantasy: that of being a powerful leader of a country in crisis. Given the real world environment in which we live today, where students routinely discuss terrorism, the taking of hostages, coup d'etats, nuclear threats, revolutions, and the systematic disenfranchisement of certain groups of people, this game is one with which players can identify. Upon being informed by the screen that the King of Strato had taken hostages as a result of lack of aid from Libros, one student exclaimed, "That's just like Iran!" The extrinsic goal in **The Discussion Generator**, amassing points, is fixed and is made compelling through the use of the visual scoreboard and the textual report on the state of the countries. We did see some evidence of emergent goals, "...goals that arise out of the interaction between a person and the environment" (Malone, p. 86). Many students wanted to find out more regarding geography and history. One student discussed the possibility of creating a similar game. Another created scenarios fashioned after the pretest, using information he had sought from recent TV news broadcasts.

Malone's framework for his Theory of Intrinsically Motivating Instruction emphasizes uncertain outcome and states that variable difficulty level, multiple level goals, hidden information and randomness are ways in which a goal can be made uncertain. While **The Discussion Generator** does not have a choice of difficulty level, the randomness of the outcome of each decision made by the players makes the uncertainness of the outcome intrinsically embedded in the program. In response to the King of Strato requesting more aid, two groups of players chose different responses to decision #2. As each group had set different initial priorities, each received different outcomes to the same decision. As they played the game a few times, students began to realize that they could not predict the actual outcome as it depended upon the priorities set and the randomness of the program itself. One student compared this game to Adventure and Odyssey. Many also began to understand that their choices did have some gross determining effect.

The Discussion Generator includes hidden information as a motivating device. At each turn, a secret advisor appears on the screen and imparts advice. He reminds players of their priorities and suggests some possibilities. This character modeled after "Deep Throat" enticed our players. Some sought to name him and give him a personality. Others questioned his motives. All looked forward to his "words".

Challenge in a game must promote rather than discourage self-esteem (Malone, 1984). The manner in which feedback is provided and the difficulty of the task can be determinates. The Generator provides non-judgmental feedback in that the scores are reported with no comment. Players can judge their own performance in terms of their own goals. One group received no points for priorities #1 and #2, but received ten points for #3 and five points for #4. One member of the group stated, "Well, we did a good job on three and four, and we weren't even trying for those!". A group which received a final score of five points, the priority partially achieved being #4, commented, "We will listen to Stephen next time. He always was against the majority decision. Maybe he knows the answers. Let's play again and see what happens!". We saw no evidence of damage to self-esteem even in groups who received very low scores. All were motivated to play again and wanted to use the game "now".

Extrinsic fantasy, as defined by Malone, is "...where the fantasy depends on the use of the skill, but not vice versa" (p. 87). He describes intrinsic fantasy as depending "...on the skill, but the skill also depends on the fantasy" (p. 87). A CAI program, such as **Math Blaster**, is a good example of extrinsic fantasy where the skill of the player in number facts determines the fantasy result of shooting down an object. **Odyssey**, an adventure type game in which a player explores a vast continent and a sea using reading and writing skills, is an example of an intrinsic fantasy. Both of these programs are used by our students.

Malone makes a strong case for intrinsic fantasy being a better instructional tool and motivator. Intrinsic fantasies help learners to apply old knowledge to new situations and improve memory through vivid images and the provocation of emotions. In this curriculum design we asked the students to role play the part of their "advisor" during the discussion. We found that (1) students were more involved in fantasy when role playing, (2) old knowledge was invoked in many situations, and (3) emotional provocation occurred, and enhanced the intensity with which the students played the game.

When first introduced to the game, the students found role playing difficult as they were unfamiliar with much of the historical data they had to read in the Resource Guide. Having to read, comprehend and interpret to others the material offered by their "advisor" required concentration and inhibited the skills required for role playing. However, once comfortable with the format and familiar with the context of the Resource Guide, the students engaged actively in their roles. The intensity of the game when they were more strongly identifying with their roles was markedly heightened. A feature of the game which enhanced this effect is that the "advisors" for each of the five moves take the same point of view. For example, if you are assigned "#1 advisor", your position for all moves may be that of a "Hawk". If you assume the role of "#2 advisor", you may have to play the role of a "Dove". As each student played the game, he/she showed more animation and familiarity with the role. An interesting twist that could be used in this conjunction, but which we did not try, would be to have players switch advisor roles at each turn. We did assign different "advisor" numbers to students each time they played a new game in order to give them the experience of defending different points of view.

Aware of research suggesting that engagement in certain types of fantasy can have a negative influence on young children (Liebert, Sprafkin and Davidson, 1982), we watched for indications of students choosing violence and aggression when deciding on their moves. We saw instead a unanimous tendency toward negotiation vs. action. Over and over again, even in the face of previous failure to produce points as a result, the students argued for and acted upon the choices including negotiation. This suggests that the students strongly identified with the situation. Discussion continually revolved around consequences of aggressive actions. "If we mobilize forces the Torans will do the same." "We can't make them (the Stratoans) think we want to take over their country!". "No! Don't let them think we will fight them!". Even in the situation where the Stratoans had taken hostages as a result of threatened military mobilization by the Librotans, the players chose to back down rather than go in to save their citizens.

While part of the tendency toward negotiation may have come from identification, the game itself is designed to present real consequences to aggressive actions, thus reinforcing the negative aspects of impulsive and aggressive decisions.

Malone's last category in his framework for his Theory of Intrinsically Motivating Instruction is curiosity. He divides this category into sensory curiosity, "...sensory stimuli of an environment", and cognitive curiosity, "...evoked by the prospect of modifying higher level cognitive structures" (p. 89).

The Discussion Generator uses both audio and visual effects to provoke curiosity. The mysterious advisor appears on the screen as a shadowed individual, each decision made invokes a screen image of the map showing the decision's effect on the three countries involved, and "beeps" precede certain requests for information. Much more could be done to enhance the sensory stimuli. The appearance of the mysterious advisor could be different each time. More graphic displays of the actions taking place in the countries could be produced. However, as this game is designed for the one computer classroom and intended to be used as a discussion generator, using the teacher as mediator, motivating students via the screen displays might inhibit design goals. If students were strongly attracted to the screen, playing the game with groups of students larger than four would be difficult, and discussion time might be discouraged as students vied for a chance to see the graphics. As the design stands now, we found that students focused on information gathering from the Resource Guide, and discussions regarding choices and consequences. While they were interested in watching the screen as their decisions were recorded, many times students allowed others to report on the results shown and did not attempt to get close enough to see for themselves. This particular feature allowed us the flexibility of playing the game with the class as a whole, facilitating our roles as mediators.

We found that more than sensory curiosity, cognitive curiosity was provoked. Malone suggests that "...the way to engage learners' curiosity is to present just enough information to make their existing knowledge e seem incomplete, inconsistent, or unparsimonious" (p. 89). This, he feels, will make them motivated to learn more. We found this to be true in using **The Discussion Generator**. The Resource Guide presents brief historical references, long enough to present the concept and short enough to hold attention. It is quite an impressive writing project. Students continually asked questions seeking more information regarding the Guide's information. "Is there still an SDS?" "Which countries are called 'imperialist' today?" They also sought information from their parents. The assignment of the oral history project at the beginning of the unit established communication and information seeking at home. Parents reported that students asked questions, not only at the initial stage of the unit, but also during it. "Where were you when the Cuban Missile Crisis happened?" asked one student at the family dinner table. Another asked her father if he believed in the Monroe Doctrine.

STUDENT RESPONSES

MOTIVATION

Evidence of student motivation were constant during the unit. The active engagement of student feedback and monitoring of the curriculum design acted as an empowering device (Brown, 1984). For the first few days of student work on the unit, we reviewed the goals of the curriculum and formally asked for opinions and examples of new skills or knowledge, emphasizing transfer and motivation. During the remainder of the unit, individual students began to report their evidence of new skill and knowledge at different times during the day. Also, as the class periods devoted to this unit began, they would begin with no teacher reminder to discuss ways in which they had used skills developed during the unit.

Some student comments recorded during the unit follow:

"I never knew it could be so hectic - making decisions."
"We have to use a bunch of different skills - all at once!"
"We had to think about consequences."
"I thought I knew how to debate!"
"We know so many new words!"

This continual discussion may have acted both as an extrinsic and intrinsic motivator. When students reported a new insight she/he was rewarded by peer and teacher approval. This was especially true at the beginning of the unit where all reporting was teacher directed within the class as a group. This extrinsic motivation may have been the impetus at the beginning. However, as the unit progressed, student reporting became more private, more spontaneous and less tied to group discussion. The level and quality of the reports changed. "I learned what 'democratize' means," reported one student during an early discussion. This same student at a later date reported, "Working with my reporting group in science reminded me of the games (**The Discussion Generator** and **The Other Side**). I tried negotiation and it almost worked!"

Discussions regarding learning styles were revealing. Students reported that they learned a great deal from the oral history project. "I talked to my Dad. He used examples which made it clear to me," reported one girl. This prompted a discussion of how much students learn from their conversations with adults. One boy reported that when he rides in a car with his dad he looks out the window and "wonders". He asks his dad questions and his dad "explains ...and explains...and explains!" This student concluded that he learns about half of what his dad thinks he has taught as a result of loosing interest at some point during the "lecture". This led to further discussion regarding learning when doing rather than when listening. The majority of students reported satisfaction with the varied materials and assignments for this unit because, "We weren't bored. Every time we did a different thing, we used something from the time before."

Another thread that was unraveled during discussions involving learning styles was that of adversity. One student reported climbing a tree and trying to back down. Being unable to do so as a result of his body angle, he knew he had to figure out a different method as there was no one around to help. He related this to his initial frustration with designing a political cartoon for his "TIME" magazine. "I knew," he said with a shake of his head, "that objectivity was important. I also knew that these kinds of cartoons are not objective. I was confused! I decided to talk to Mrs. Barrett. It was amazing how quickly she cleared the muddle!" Another student revealed his consternation while attempting to write his science/medicine section for his "TIME". "I wondered how prehistoric man could even

relate to science and medicine! We are so advanced now...'no way', I said. But it had to be done. So...I asked around, you know, everyone I saw. Finally, I got the message that all you guys gave me...wheels and fire!". The class discussion then focused upon challenge as a motivational tool and the importance of this element in their learning experiences. One girl excitedly said, "That's like that poem we read...You know, the Grook." She then recalled a poem of Piet Heim's which reads, "A problem worthy of attack, proves its worth by hitting back", another example of using old knowledge to reinforce new ideas.

Some discussion was around choice. One student said, "I enjoy learning more on my own will." When encouraged to elaborate, she related that when the choice is hers, and she is not under the pressure of time or grades, she enjoys the learning in which she chooses to engage. As an example, she discussed the videos available with the unit. "Knowing that I could choose any or none of the videos caused me to look at most of them. I speeded forward and back at my own will. I learned a lot from them and chose to look at more than I would have if you had told me to do it." Another student discussed her brother's interest in a particular type of music. "He plays it all the time. I listen when I want to, and when he and I are riding in the car. I don't believe how much I know when he asks me questions!" This suggests that choice within structure is an important motivational tool.

As can be expected, there was discussion concerning the motivational effects of interactive media. There was unanimous agreement that **The Discussion Generator**, the videos and the film, "The Mouse That Roared", were "awesome". One student compared the videos and film in this unit to a film from which she had learned about disease in children. "You know," she said, "that film called 'Alex: The Life of a Child'? It seemed so real. I really understood what they meant. Cystic Fibrosis is really hard! That's what I felt when I saw the REAL Hitler on the video. Somehow I finally believed that there was realism in that time." Our observations suggest that structure is an important part of the curriculum, and that choice should be available at as many levels as possible.

As the unit contained opportunities to work individually, the creation of the "TIME" magazine, individual meetings with teacher/advisor, and within a group, using the software, choosing enrichment projects and class discussions, we had the opportunity to compare motivation in both circumstances. It is important to emphasize again that these students had previous experience with a collaborative game, **The Other Side**. They referred to their past experiences with this software often. Our observations suggest that their ease with collaboration during sessions with **The Discussion Generator** was as a result of work with **The Other Side**. When first introduced to **The Other Side**, many students had difficulty with collaboration. Leadership within groups changed daily and the quieter students had some difficulty participating. As they gained more experience with the game, collaboration became an articulated goal. In approaching **The Discussion Generator**, most students used the skills they had learned from earlier collaboration.

From the beginning of this unit, all students reported enthusiasm and pleasure regarding small group use of **The Discussion Generator**. Repeated requests were made to use the software during other parts of the day. Collaborative skills were present and intensified during the unit. Leadership within the groups seemed not to be an issue as consensus was the goal. What emerged was a desire on the parts of the students to convince others through discussion that their "expert's" advice was correct. "I was convincing," reported one student, "and it was because I knew the history and they didn't." As each game of **The Discussion Generator** ended, groups reviewed their decisions, "If only we had...", "What if...", and "Next time let's try..." were heard often. We found that **The Discussion Generator** encouraged students to look at the strategies of others. They asked other groups to relate their list of initial priorities and to catalogue their moves.

While the initial motivation to play the game may have been extrinsic, the highest score, the chance to play with a computer, we saw more evidence of intrinsic motivation in enthusiasm for "opportunities for cooperation, competition and recognition" (Chen, p. 51).

Playing the game as a class unit did not engender the same enthusiasm. Students reported that while it was "fun", everyone did not get the chance to "argue and convince others". In the time allowed, all 21 students did not have equal opportunity to express opinions. One student said, "If the goal is to become better thinkers and arguers so that we have more knowledge, we have to be able to talk." We suggest that while group work at computers may promote learning as students collaborate to overcome problems (Chen, 1984), the size of a group is of prime importance, especially when promoting skills involving cooperation and collaboration, and rational argument.

Class discussions regarding structure of the unit and the reporting of goals met were enthusiastically received. However, because the information given and solicited did not call for rational arguments presented through lengthy discussion, a comparison of class work with **The Discussion Generator** cannot be made.

The "TIME" magazine assignment emphasized opportunity for students to discover individual interests and talents (Gardiner, 1985). The assignment included choosing, researching, expository and editorial writing, drawing, designing and creating.

Choosing a time period was especially enticing to them. A great deal of discussion among students regarding possible choices was observed. Each student had recently completed the writing of a 2500 word research report, and it was interesting to note that many students chose to develop their magazine around the knowledge they had gained from their report, thus building on pre-existing knowledge. One student who had done her report on the Holocaust chose a five year period during World War II. Another who had researched an authoress chose a five year period in history when the same authoress would have been beginning her career. The challenge of doing a magazine from pre-historic times was assumed by one student. Another chose to begin with his birthdate. Others chose times in their parents' or grandparents' lives, as they recalled the earlier oral history assignments both in this unit and in others designed earlier in the year.

The actual creation of the magazines involved practice of a variety of skills. As could be expected, depending on the interests and talents of the individual students, each magazine showed different places of expertise. Motivation for the project as a whole was evident. However, students invariably met with frustration as they approached the parts of the assignment that were hard for them. We continually monitored progress and problems both by meetings with individuals and observations of work in progress. The teacher/advisor was important here in the sustaining of motivation. Those students whose visual skills were weak were aided by the Head of the Art Department. Others needed help with research and gained it through the Librarian. Those whose science/medicine section seemed impossible were aided by the science teacher. These one-on-one tutorials were essential for avoiding destructive frustration leading to damaging self-image.

TRANSFER

Our attempts to document transfer were unsuccessful. This was partly due to the fact that the unit is still in its final stages. In order to document High Road Transfer (Perkins, lecture), we will need to continue to monitor the experience of students in other areas of the curriculum. Abstraction and conceptual action (Perkins, lecture) will have to be observed in the context of new situations.

One of the places that we had hoped to test for High Road Transfer was in the choice and execution of the voluntary end-unit activities. The choices for this part of the unit have been made, but the final products are not due for another two weeks. What is interesting is that all 21 students chose to embark on an "enrichment" project. The majority chose the Video production. We are looking forward to observing and mediating in this attempt. We will be looking for choices of news items, objective reporting, and an understanding of the complexities of the issues.

We were able to observe some evidences of transfer in school activities unrelated to the academic agenda. The sixth grade at this particular school is the "graduating class". As such, they have class elections and wield peer power within the school. Class meetings are used as a way in which to allow them to experience leadership and followership without adult intervention. This particular class had, before this unit, acted as past classes in that class meetings were chaos. Following this unit they held a class meeting which we observed. The results were quite different. There seemed to be more willingness to listen to the views of others. "Wait!", yelled one student as the meeting began to disintegrate. "Let's listen to Stephen. Remember he had the ideas in the game." Silence followed. Rational discussion commenced and a plan of action was formulated, regarding a fund raising activity, quite different from previous years. Stephen's ideas did not prevail, but the mere fact that listening occurred changed the direction of the meeting. Jobs were assigned after a discussion of the goals of the project. Throughout the discussion there were numbers of incidents where students verbally pondered the ramifications of their actions on the various school constituencies. Debating skills were observed as well as the tendency to listen to the opinions of others.

Another instance occurred as the class was discussing an upcoming week long survival trip. Past classes have vociferously expressed great negative reactions regarding the formation of the group into two sections which would not communicate for five days while on this five day trip. This issue has never been rationally resolved until a culminating discussion upon each class return from the experience. As the annual discussion progressed this year, rational arguments and understanding of other issues prevailed. "I guess that there is no way that they can put 21 kids together and make it work," said one student. "If we are to learn anything,'" said another, "We can't be with only the people we like." A classmate responded, "What would that do? We would just be doing the same thing...not getting the opinions of those we don't know." We asked them where this attitude originated. One student said, "You already know!" The Discussion Unit." The general tone and means of dealing with the situation was markedly different from previous years.

CONCLUSION

While we are quite aware of the primitive research devices used, we feel that the unit was successful in terms of results seen from explicit instruction, design from cognitive research and the inclusion of the software that we chose, **The Discussion Generator**. Metacognitive research suggests that a "major component of effective learning involves planning, comprehensive monitoring, and selecting appropriate strategies, as well as effective management of stress and time (Wolfson, 1986). Our experience with this unit verified this approach. We look forward to continuing to mediate and monitor the conclusion of the unit and in the planning of others with the same goals in mind.

APPENDIX B

VOCABULARY LIST OF ORAL HISTORY PROJECT

ENTEBBE
BOXER REBELLION
FIDEL CASTRO
HENRY CLAY
IMPERIALISM
KOREAN WAR
CHIANG KAI-SHEK
CUBAN MISSILE CRISIS
BAY OF PIGS
MONROE DOCTRINE
DOMINO THEORY
SANCTION
S.D.S.
YELLOW PRESS
SIT IN
INTERVENTION
HAWK
DOVE
GRENADA
NURENBURG

APPENDIX C

LIST OF COUNTRIES LOCATED AND DISCUSSED AS A PREACTIVITY

AFGHANISTAN
ANGOLA
ATLANTIC/PACIFIC OCEANS
AUSTRIA
CAMBODIA
CANADA
CENTRAL AMERICA
CHILE
CHINA
COLOMBIA
CUBA
DOMINICAN REPUBLIC
EGYPT
ENGLAND (BRITAIN)
FRANCE
GERMANY
GRENADA
GUIANA
HAWAII
IRAN
ISREAL
JAPAN
KOREA
LAOS
LATIN AMERICA
LEBANON
MALTA
MEXICO
MIDDLE EAST
NICARAGUA
PANAMA
PHILIPPINES
SOUTH AFRICA
SPAIN
TAIWAN
UGANDA
VENEZUELA
VIET NAM
U.S.A.
U.S.S.R.

APPENDIX F

Source used for examples of yellow journalism and point of view in reporting:

Harnadek, A.\Critical Thinking Book 1\California Midwest Publications Co., Inc., 1976.

Chapter 5 - PROPAGANDA TECHNIQUES

5.1 Introduction
5.2 "Bandwagon"
5.3 Repetition
5.4 "Transfer"
5.5 "Testimonial"
5.6 "Exigency"
5.7 "Free" and "Bargain"
5.8 "Glittering Generality"
5.9 Innuendo
5.10 "Name-Calling"
5.11 Miscellaneous Techniques

Chapter 8 - LEARNING TO BE OPEN-MINDED

8.1 Introduction
8.2 Looking at Two Sides of an Issue
8.3 Recognizing Issues and Supporting Points
8.4 Anticipation Arguments for **The Other Side**
8.5 Finding Arguments of Your Own
8.6 Debating

This is a student workbook which can be used as a source or given to each student. We had one copy for each student.

APPENDIX F - LIST OF VIDEOS

SERIES: "20th Century History"

NUMBER	TITLE
1.	Make Germany Pay
2.	Boom and Bust
3.	Hitler's Germany 1933-1936
4.	Roosevelt & The New Deal
5.	Why Appeasement?
6.	Britain Alone
7.	Stalin & the Modernization of Russia
8.	Pearl Harbor to Hiroshima
9.	India-The Brightest Jewel
10.	Cold War Confrontation
11.	The Road to Berlin
12.	Mr. Kennedy & Mr. Krushchev
13.	One Man's Revolution - Mao Tse-Tung
14.	Isreal and the Arab States

Available from Massachusetts Educational Television, a division of the Massachusetts Department of Education, 27 Cedar Street, Wellesley, MA 02181. (617) 431-7013. Purchaser provided two blank tapes and paid $40.00 for copying fee.

BIBLIOGRAPHY

Brown, J.S., "Process Versus Product: A Perspective On Tools for Communal and Informal Electronic Learning." In \Children and Microcomputers\ Edited by M. Chen and W. Paisley. California: Sage Publications, Inc., 1985.

Chen, M., "A Macro-Focus on Microcomputers". In \Children and Microcomputers\ Edited by M. Chen and W. Paisley. California: Sage Publications, Inc., 1985.

Jones, B. F., "Quality and Equality Through Cognitive Instruction." In "Educational Leadership", Col. 43, No. 7, April, 1986.

Liebert, R., Sprafkin, J., Davidson, E., \The Early Window\ New York: Pergamon Press, 1982.

Malone, T.W., "Toward a Theory of Intrinsically Motivating Instruction." In \Instructional Software: Principles and Perspectives for Design and Use.\ Edited by D. Walker and R. Hess. California: Wadsworth, Inc., 1984

Walters, J., and Gardiner, H., "The Development and Education of Intelligences." In \Essays On The Intellect\ Edited by F. Link. Virginia: Association for Supervision and Curriculum Development, 1985.

Wolfson, B., "Psychological Theory and Curricular Thinking." In \Current Thought on Curriculum\ Virginia: Association for Supervision and Curriculum Development, 1986.

COLLECTING EYE MOVEMENT DATA USING AN APPLE IIE
A CASE STUDY

By: Bud Wellington, University of South Florida, Tampa, FL

INTRODUCTION

This paper presents excerpts from research in progress designed to explore eye movements during the solution of geometric analogies. Methodologies developed through previous eye movement research are applied within the context of the information processing paradigm to investigate the solution of geometric analogies. The intent of *this* paper is to demonstrate the potential of computer facilitated data collection by showing its application in a specific instance. The material included here provides a brief overview of the problem being investigated, and then focuses on the more technical aspects of interfacing, data collection, storage and reduction. The paper does not address details of the research design per se.

EDUCATIONAL RELEVANCE

Several researchers have applied eye movement methodologies to educational questions. Raynor (1984) and McKonkie (1976) have used eye movement analysis to investigate the dynamics of reading. Pavlidis (1985) has demonstrated significant differences in the eye movement patterns of normal and dislexic readers. Boersma and Wilton (1974) and Vurpillot (1968) couple eye movement analysis with Piagetian theory to investigate developmental differences in search and problem solving activities. Boersma and Muir (1975) investigate discriminate learning in normal and retardate populations by using eye movement analysis in the context of the Behaviorist paradigm. Yet, the potential for eye movement analysis to enhance educator's understanding of problem solving processes and related instructional strategies has remained largely uninvestigated.

This study investigates three areas of interest to educators: 1) eye movement methodology, 2) individual differences among subjects, and 3) theory of analogy solving. Eye movement analysis as a method of measurement is important because it is the single most visible and therefore quantifiable indicator of cognitive activity. Individual differences are important because they play such a vital part, especially in special education, in the development and implementation of differentiated curriculum. Knowledge about the process of analogy solution is important first because the use of analogy as both a teaching technique and, more recently as curriculum content. Second, because analogy is said to be a strong indicator of intelligence. Further, a knowledge of analogy construction (as distinct from solution) can aid the educator in developing appropriate instructional materials.

EYE MOVEMENT METHODOLOGY

The ability of computers to interface with the "real" world has opened new vistas in eye movement research. Consider the three optimal conditions for data collection put forth by Sheena and Borah (1981). The first principle suggests there should be maximum physical and psychological freedom for the subject; data collection should be non-intrusive. A second goal is to collect the maximum amount and type of data with highest possible accuracy. Finally, data collection techniques should accommodate a wide variety in population parameters (eg. eye color, eye shape, visual accuity). This real world interfacing capacity of computers coupled with their inherent speed and accuracy, not only facilitates the researcher's ability to maintain ideal conditions, but opens the door to the collection and analysis of data on a scale which seemed visionary only a few years ago.

Information from the visual environment is selected and processed selectively (Cohen 1981). The length of time a person fixates on an object may be used to infer mental processes. Further, the number of fixations is more significant in recall than the duration of fixations Loftus (1972). The pattern of successive fixations provides an observable indication of selective processing. A classic study by Yarbus (1967) in which subjects were asked to view a picture under different task conditions illustrates the point well: analysis revealed distinct scanpaths for each task. Cognitive strategies are needed to determine the location of a fixation and to encode the information once it is located (Russo, 1978). Almost all saccadic eye movements are controlled by cognitive processes. It has been said that the eyes are the windows of the soul. More recently, and more practically, it has been shown that the eyes are windows to the mind. Investigators have established a clear link between cognitive process and eye movement. Eye movement is the single most visible and therefore measurable indicator of cognitive activity. Coupled with cognitive psychology, specifically the information processing paradigm, the analysis of eye movement data provides significant insight into understanding fundamental cognitive processes.

THE ANALOGY

In terms of both quantity and quality, research grounded in the information processing model yields greatest insight into the process components of problem solving. Several investigators represent theories of analogical reasoning. Spearman's (1927) discussion of the nature of intelligence is essentially an information processing model. Shalom and Schlesinger (1972) extend the Spearman theory. Reitman (1965) and Evans (1968) have developed computer programs to solve verbal and geometric analogies.

The bulk of theory building and research based specifically on the analogy rests with Sternberg (1977, 1985), who has developed a componential theory of analogical reasoning and a plausible, detailed model of analogy solving. Sternberg's componential theory clearly delineates the process components of analogical reasoning and their relationship to each other. Sternberg's (1977) contribution to the field is immense; the theory and resulting models provide the foundation and methodology which has defined the field. Sternberg does not remain unchallenged, however. Mulholland, Pellegrino and Glaser (1980), while maintaining the framework of the Sternberg model, contribute to the theory of analogy solving by manipulating the structure of analogy items. The resulting Mulholland model details the effects of item construction and difficulty and provides the best fit to data.

DEFINITIONS

Analogy

An analogy consists of four term relationship of the form A is to B as C is to D (A:B :: C:D). Consistency must be maintained for both the lower order (A:B and C:D) relationship and the higher order (A:C and B:D) relationship. Analogies may be verbal, numeric, or figural and are typically presented in one of two forms: 1) true-false in which the task is to determine whether the given analogy is true or false, and 2) forced choice, in which the task is to choose one of several alternatives which best completes the analogy. Geometric analogies are a subclass of figural analogies. This research employs multiple forced choice geometric analogies from the Cognitive Abilities Test (Thorndike and Hagan, 1971) of the form:

For consistency within this study, item refers to an analogy problem; term refers to parts within the item (e.g. in the analogy A:B as C:D, "A" is the first term, "B" is the second term, etc.); element refers to the geometric shapes (e.g. square, cross, line, circle) which comprise the terms. Transformation refers to changes or operations (e.g. elongation, rotation, shading) performed upon elements to create the analogy.

Sternberg's (1977) theory of analogy solution comprises five mandatory and one optional processes. Encoding is the process whereby attributes and values of each term in the analogy are identified and stored in working memory. Inference is the process whereby the rule (or rules) relating A to B is discovered. Mapping is the process whereby the rule (or rules) relating A to C is discovered. Application is the process whereby the inference and mapping rules are used to generate an ideal D solution and the solution is evaluated against alternatives. Justification (optional) is the process whereby the validity of operational decisions is rechecked. Preparation-response is the process whereby the solution process begun, monitored and a response emitted. Again, a complete discussion of the solution process is presented in the evaluation of literature.

Eye movement patterns

Analysis of eye movement patterns typically involves consideration of the fixation, or stopping of the eye at a point in the visual field, and the saccade, or motion of the eye as it travels from one fixation to the next. A scanpath (Stark and Ellis, 1981) is the sequence of fixations and saccades executed during the solution during task completion. It is the eye movement component of a subject's response to an item. For this study, scanpath and eye movement record are used synonymously. Two additional eye movement patterns are also employed in this study: 1) paired comparisons are defined as a sequence of fixations in which the first and third fixations are to the same term, and 2) runs, which are defined as a sequence of non-repeating fixations over three or more terms.

Instrumentation

The Geometric Analogies subtest of the Nonverbal Battery (Test 2, Level H) of the Cognitive Abilities Test (Thorndike and Hagan, 1971) was chosen for both the pretest screening and the eye movement phase of the study.

A set of 35mm black and white slides were produced from the 25 items of the Geometric Analogies subtest (Form 3, Level H) of the Cognitive Abilities Test. Items were reproduced exactly as shown in the test manual with the exception that instead of response choices being labeled by letters above each choice, the slides presented response choices being labeled with numbers below each choice. Two additional slides were produced to serve as sample items and a set of calibration slides was also prepared so that during the collection of eye movement data each item could be preceded by calibration. The calibration points are important in later mapping the eye movement data back onto the stimulus material.

Apparatus

A Kodak Ectographic carousel slide projector with autofocus zoom lense and wire remote control was used to project the slides onto a 6" by 12" rear projection screen. Subjects were seated at a distance of 23.5" from the front of the screen, resulting in a 27 degree field of view, which is equal to the normal field of view when the test is administered under standard conditions (Mullholland, 1980), and nearly maximizes the (30 degree) range of the eye movement monitor.

Eye movements were monitored by an Applied Science Laboratories Model 210 Eye Movement Monitor and routed via a California Computer Systems Model 7720 Parallel Interface to an Apple IIe computer where the data was recorded and stored for future analysis. The computer was equipped with an Applied Engineering Ram Works II 512K memory expansion card, and an Applied Engineering Timemaster H.O. time card to enable millisecond timing. The subject response button was wired to pin 2 (Pushbutton 0) of the Apple game port. Finally, in order to ensure consistent operation of the computer during test administration, the remote control unit for the slide projector was mounted on the keyboard, and those keys which would be used for the calibration procedure, item starting and change fo slide were designated and clearly labeled.

Software to drive the interface board, access the time card, collect and store the data was written and compiled using the S-C Macro Assembler 2.0 (Sander-Cederlog, 1982). The program captures and records:

1) Subject and item identification data;
2) Calibration points;
3) Start and end time for each item;
4) Eye position at 20 millisecond intervals;
5) Eye position at time of response.

Data Reduction

As detailed in the discussion of apparatus, eye movement data was collected at a rate of 50 samples per second. The accumulated data was then written to disk for later reduction and analysis. As typical of eye movement studies, analysis began by reducing the raw data for each item. The reduction process involved three aspects:

1) Differentiate fixations from saccades; determine the duration of fixations and length of saccades;
2) Determine the locations of fixations and saccades relative to the stimulus material;
3) Calculate appropriate means; count runs and comparisons.

Two data reduction programs were written in BASIC to perform the desired tasks. The algorithm to determine fixations was constructed using the position-variance method described by Anliker (1976) and by Widdel (1984). Two sliding "windows", each comprising five consecutive samples were established such that when the first window encompassed data pieces 1 through 5, the second window would encompass data pieces 2 through 6. The mean and variance of each window was calculated and compared. By hand checking algorithm output with a printout of raw data, it was empirically determined that if the variance in both windows was greater than 1.0, a saccade had started. If the variance in either window was below the 1.0 threshold, a fixation was indicated. By sliding both windows ahead one data point, then recalculating the means and variances, a running account of fixations and saccades was achieved. Counters in the program kept track of the durations and "raw" locations of fixations and the lengths of saccades.

The algorithm for calculating the locations of fixations and saccades relative to stimulus material was developed empirically by measuring distances, calculating proportions within the stimulus material relative to the calibration points. The algorithm established eight zones, one zone for each term of the analogies. As was indicated in the pilot study, some fixations were found to lie outside collected calibration points. In those cases, a revised calibration point was calculated based on the mean distance from the leftmost fixation and left calibration point for items where all fixations fell within the original calibration points. The resulting zones were hand checked against printouts of the raw data to verify their viability. After zones were calculated, the raw locations held by the first algorithm were transformed into zone locations.

The calculation of means and variances of fixations and saccades, and of response times was straightforward and requires no comment. Output from the first data reduction program was written to disk as a text file for each item and comprised: number of fixations, position of each fixation relative to stimulus material (zone), response time, duration of fixation #1, mean duration of fixations in milliseconds, variance of fixations, mean length of saccades in degrees, and variance of saccades.

The second data reduction program, CZ TO ENTER, read the text files output by CZ FILE back into an array and performed three functions. First, multiple sequential fixations within the same zone were condensed to a single fixation so that runs and comparisons could be counted. Second, the new zone list array was examined for runs. The run detection algorithm created a sliding window in the array three data pieces wide, evaluated the pieces by checking to see if each successive piece was one greater (or less) than the preceding piece. A counter kept track of the number of runs detected. Finally, the comparison detection algorithm created a sliding window three data pieces wide and evaluated by checking to see if alternating data pieces were equal. A counter kept track of the number of comparisons detected.

CONCLUSION

At this point in the conduct of the study, the reduced data is ready for analysis. In the present study, the design is causal-comparative, with an exploratory approach in the analysis of data. The two groups formed on the basis of performance on the Geometric Analogies subtest of the Cognitive Abilities Test (Nonverbal battery, Form 1, Test 2, Level H: Thorndike and Hagan, 1971) will be compared on selected eye movement variables. Since a full discussion of the design and results of the study are beyond the scope of this paper, the interested reader is urged to contact the author for further information.

SELECTED REFERENCES
Eye Movement, Cognitive Process and Problem Solving

Anliker, J. (1976). Eye movements: online measurement, analysis and control. In R.A. Monty & J.W. Senders (Eds.), Eye movements and psychological processes. Hillsdale, N.J.: Lawrence Erlbaum Associates.

Boersma, F.J. and Muir W. (1975). Eye movements and information processing in mentally retarded children. Rotterdam University Press.

Boersma, F.J. and Wilton, K. (1974). Eye movements and conservation acceleration. Journal of Experimental Child Psychology, 17, 49-60.

Carpenter, Patricia A. and Just, Marcel A. (1978). Eye fixations during mental rotation. In J.W. Senders, D.F. Fisher and R.A. Monty (Eds.) Eye Movements and the Higher Psychological Functions. Hillsdale, New Jersey: Lawrence Erlbaum.

Cohen, K.M. (1981), The development of strategies of visual search. In D.F. Fisher, R.A. Monty & J.W. Senders (Eds.), Eye movements: cognition and visual perception. Hillsdale, New Jersey: Lawrence Erlbaum.

Edenart-Pepe, R.L. (1983). Responding in eye movement strategies used to solve the adolescent/adult matching familiar figures test. Ann Arbor, MI: University Microfilms International.

Evans, T. G. (1968). Program for the solution of a class of geometric-analogy intelligence-test questions. In M. Minsky (Ed.), Semantic Information Processing. Cambridge, Mass: MIT Press.

Loftus, G.R. (1972). Eye fixations and recognition memory for pictures. Cognitive Psychology 3, 525-551.

Mackworth, N. H. (1976). Ways of recording line of sight. In R.A. Monty & J. W. Senders (Eds.), Eye movements and psychological processes. Hillsdale, N.J.: Lawrence Erlbaum Associates.

Mackworth, N.H. (1976a). Stimulus density limits the useful field of view. In R.A. Monty & J.W. Senders (Eds.), Eye Movements and Psychological Processes. Hillsdale, N.J.: Lawrence Erlbaum Associates.

McConkie, G.W. (1976). The use of eye-movement data in determining the perceptual span in reading. In R.A. Monty & J.W. Senders (Eds.), Eye movements and psychological processes. Hillsdale, N.J.: Lawrence Erlbaum Associates.

Mulholland, T.M., Pellegrino, J.W. and Glaser, R. (1980). Components of geometric analogy solution. Cognitive Psychology. 12, 252-284.

Pavlidis, G. (1985). Eye movements in dyslexia: their diagnostic significance. Journal of Learning Disabilities, 18, 42-50.

Raynor, K. (1984). Eye Movements and Reading. New York: Academic Press.

Reitman, W. (1965). Cognition and Thought. New York: Wiley.

Rowe, H.A.H. (1985). Problem Solving and Intelligence. Hillsdale: N.J.: Lawrence Erlbaum Associates.

Russo, J.E. (1978). Adaptation of cognitive process to the eye movement system. In J.W. Senders, D.F. Fisher & R.A. Monty (Eds.), Eye movements and the higher/psychological functions. Hillsdale, N.J.: Lawrence Erlbaum Associates.

Sander-Cederlof, R. (1982). S-C macro assembler. Dallas, TX: S-C Software Corporation.

Shalom, H. and Schlesinger, I.M. (1972). Analogical thinking: A conceptual analysis of analogy tests. In R. Feurstein, I.M. Schlesinger, H. Shalom & H. Narrol. Studies in cognitive Modifiability, Report l, Vol. II. Jerusalem: Hadassah Wizzo Canada Research Institute.

Sheena, D. & Borah, J. (1981). Compensation for some second order effects to improve eye position measurements. In D. F. Fisher, R.A. Monty & J.W. Senders (Eds.), Eye movements: cognition and visual perception. Hillsdale, N.J.: Lawrence Erlbaum Associates.

Spearman, C. (1927). The Abilities of Man. New York; Macmillan.

Stark, L. and Ellis, S.R. (1981). Scan paths revisited: cognitive models direct active looking. In D.F. Fisher, R.A. Monty & J.W. Senders (Eds.), Eye movements cognition and visual perception. Hillsdale, N.J.: Lawrence Erlbaum Associates.

Sternberg, R.J. (1977). Intelligence, information processing and analogical reasoning. Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Sternberg, R.J. (1985). Human abilities: an information processing approach. New York: W.H. Freeman and Co.

Thorndike, R.L. and Hagan, E. (1971). Cognitive abilities test. Boston: Houghton-Mifflin.

Vurpillot, E. (1968). The development of scanning strategies and their relationship to visual differentiation. Journal of Experimental Child Psychology, 6, 632-650.

Widdel, H. (1984). Operational problems in analyzing eye movements. In Gale, A. and Johnson, F., (Eds.). Theoretical and Applied Aspects of Eye Movement Research. North Holland: Elsevier Science Publishers.

Yarbus, A.L. (1967). Eye Movements and Vision. New York: Plenum.

REAL-TIME COMPUTER CONFERENCING POTENTIAL FOR EDUCATION

By: Bud Wellington, University of South Florida, Tampa, FL

INTRODUCTION

Two types of computer conferencing are commonly encountered. In *message based* conferencing, a topic is developed as a series of one or two paragraph notes. Participants are free to read and add to the discussion at their convenience; they are *not* online at the same time. With realtime conferencing, however, all participants *are* online simultaneously. Discussion develops rapidly and tends to be less structured than message based conferencing; interaction is immediate and fluid. This paper explores the world of realtime conferencing.

SCENE 1: CAN WE TALK?

I've been involved with online communications for about three years now. As an educator, I sense a potential, a power, an opening, a *something* about computer mediated communications that I can't quite articulate. As educators we need to start talking about it; to start formulating the questions; to start thinking about the potential and implications that online communications has for teaching and learning.

Look at this:

(AAA) Hey - BBB - !! How are ya?
(BBB) I'm well enough. How bout you?
(AAA) Ya! Good way to put it, "well enough" .. me,too.
(BBB) I have some bad news tho.l My mom died last week.
(AAA) I am sorry to hear that, - BBB -. Do you want to talk in private for a few mins.?
(BBB) Did you have anything you wanted to talk about? I think I'm doing pretty well here. She fell asleep reading and then died in her sleep. So, she was doing something she loved.
(CCC) Hi all. What's the topic tonight?
(BBB) Well, - CCC -, My mother died last Tuesday.
(CCC) - BBB - .. I am sorry to hear that.. was it sudden?
(AAA) Nothing special, - AAA - I just want you to know I am available to chat ANYTIME you want.
(BBB) thanks so much -- Yes, - CCC - -- totally unexpected.
(CCC) do you have family support near by?
(BBB) Yup, my dad's here and brother's nearby and I have all the aunts and cousins in the world.
(CCC) it is important to have family and friends close at a time like this.
(BBB) I know, and I do.
(CCC) My Mom died 11 years ago, last Saturday. I still miss her.
(BBB) Boy do I ever miss my mom. I just wish that she could have stuck around long enough to see me graduate and go to college.
(CCC) Don't ever stop missing her..just don't let it stop you from living your own life to its fullest.
(BBB) I will never stop missing her.
(CCC) how old are you?
(BBB) 16

(CCC) - BBB -.. just remember we are always here to help..if you need someone to talk with.
(BBB) Thank you
(AAA) I REALLY hope you will keep in touch with this group, - BBB -
(CCC) I am on most evenings.. and you can always e-mail me, - AAA - .. or leave a posting in the Notice section.
(BBB) But right now I have to to, because I have beaucoup de travail and I was gone from school for 6 days.
(BBB) I will, - AAA -. I love these convo's!
(CCC) - BBB - ... our thought are with you ... Hug.
(AAA) Good night, - BBB - ... HUG.

As an individual, my humanity is touched by that conversation. As an educator, I can not avoid thinking that there is something important here. Clearly, this is not the stuff of the classroom. It is not academic, not planned, not in the book or by the book. It is not *schooling* ; it is *education* . It is real life. Will realtime computer conferencing have much of an impact on the classroom? Probably not. Will realtime computer conferencing influence our thinking about education? Most certainly. The distinction is important.

SCENCE 2: WHO ARE THESE PEOPLE?

In face to face conversations typical of everyday experience, information is transmitted in auditory and visual channels. The auditory *content* of the message and the auditory *delivery* of the message both provide cues to meaning. Further meaning is derived via visual channels: facial expression, eye contact, body language and physical appearance all contribute to the impact and meaning of the message. In realtime computer conferencing, however, none of the familiar cues is available. As a result, a style and jargon has emerged to compensate for the lacking cues. Fore example, one often encounters the question: "MORF?", meaning, "Are you **M**ale **OR** **F**emale?". Cartoon language often indicates mood, intent or action: <grin>, <oh no!>, <opening another beer>.

Anonymity prevails. Consider this conversation:

(JJJ) My personal sense of the medium is that it is VERY powerful and will be used more and more for counselling.
(XXX) what type of counselling?
(JJJ) personal... probably not heavy duty analysis, but the relatively minor stuff...
(JJJ) help dealing with the little crises we all go through.
(JJJ) outlet, venting, having an eye that will listen.
(YYY) hahahaha
(JJJ) <grin>
(JJJ) . think there is some potential here for role playing ..
(XXX) I have to admit... computing like this has opened me up... I used to be an introvert... while I have been opening up a long time.. this added to it.
(JJJ) yes, it has for me, too!
(XXX) It has helped to raise my self esteem.
(JJJ) or for letting out repressed personnae.
(*OLLIE NORTH) I refuse to answer under the grounds that it may tend to incriminate me.
(ZZZ) <cough, cough>
(ZZZ) Any repressed personnae in here?
(JJJ) have any of you encountered roll players online?
(YYY) That is a part of behaving in a civilized manner.
(ZZZ) That reminds me. Have to get my pants repressed.
(XXX) hahaha, ZZZ

(JJJ) hahahaha
(XXX) JJJ.. I can't tell.
(JJJ) I guess that's IT ...we can't tell!!

Indeed, one *can't* tell! Yet, that is part of the intrigue, part of the power of the medium. As educators we need to understand how anonymity effects communication. We need to consider the possibilities.

SCENE 3: ANYONE GOT A QUARTER?

The medium is unique, and lends itself to the creation of new forms of humor and play. Here is a "conversation" that almost got away.

```
(CCC) --------- * --- <BORED!>
(DDD)                ^
(BBB)   ^
(BBB)      ^
(DDD)           ^
(BBB)   ^
(CCC) gonna shoot, AAA?
(DDD)         ^
(BBB)  ^^^^^^^^
(DDD)         ^
(CCC) hahahahaha
(BBB) shot gun
(CCC) --------- * -------
(DDD) ---------- * -
(CCC)          ^
(DDD)            ^
(DDD) can you see someone walking in on us now?
(AAA) hahahaha - who made up this game?
(BBB) We did, just now
(BBB) still doing it too
(DDD) see? brilliant minds at work!
(CCC) go ahead, AAA .. give us a target.
(AAA) ---------------------- * ---------
(DDD)                  ^
(BBB)               ^
(DDD)                        ^
(AAA)                           !
(BBB)       ^^^
(CCC) ahhhhhh ... a bomb!
(BBB) ohnonono
(BBB)               ^^
(AAA)                     !
(CCC)                        <^>
```

Minds at work, minds at play. As an educator I *sense* that this is an important conversation. I see an impressive display of spontaneous creativity. I wonder about creative problem solving, about "thinking skills", about playfulness and flexibility in learning. I wonder about schooling and about *education*..

SCENE 4: IS ANYBODY HERE?

A network can be viewed as a data structure in which people are "nodes" and the connecting lines are relationships. A network is the antithesis of a hierarchy. Direct connections replace chains of command; relationships replace roles. Electronic proximity overpowers physical distance and our sense of neighborhood expands.

(RRR) You know, except for some of the hardships they endured...I would love to go back in time to those bygone days!
(EEE) and give up your PC, RRR?
(EEE) I don't think I would want to go back..maybe for a look only..but I like what I live now
(RRR) But the slow pace, the caring of one for another that it took back then... the family involvement...etc...
(EEE) It would be great if that involvement could be rekindled these days... there is so much to offer each other now
(EEE) JJJ...amen to that
(HHH) It seems that there is a human need for involvement with others...
(HHH) I suspect just being online here fills some of that for most of us..
(HHH) it certainly does for me.
(RRR) Yea, but sitting on the porch swing, talking to your neighbors, etc... Nowadays, who knows who your neighbor is anymore...
(JJJ) There is...and knowing your own history helps too..
(HHH) We have electronic neighbors, RRR <grin>
(HHH) <pulling up rocking chair .. moving computer to the porch>
(RRR) Thank God for NETWORK and the people it has brought into my life... hehehehe...I like that HHH!
(JJJ) True RRR, You may not know who your neighbor is, but you probably know the people you work with.
(EEE) good point JJJ..different circle of people around these days
(RRR) Well, in my case...I don't work...most of time is spent here at home... Would be mighty boring without telecommunications...Couldn't be a Ham Operation..Can't understand the Morse Code...
(HHH) yes, I think we are defining our "neighborhoods" differently now.
(RRR) But you know how we all need to have someone to call friend....
(RRR) Well, you took the words right off of my fingertips, HHH!
(JJJ) Once the cost gets down where more people can afford it, there's going to be a real explosion.

In fact there's *already* a real explosion. Like many, I know my electronic neighbors better than my physical neighbors. I live in an electronic community. As an educator, I must consider the notion that electronic communities may well develop electronic schools. I must wonder about the future of the classroom, and about the future of site-bound schooling.

CONCLUSION:

Computer mediated communication expands our personal and social space. It extends our intellectual and emotional reach at a greater speed and to a greater distance than any previous medium. And *this* is only the beginning; the technology is still young. There is a *world* of computer mediated communication growing, forming, and establishing itself as a reality in the lives of the children, parents and colleagues we see every day. Educators ought to have a place in that world.

SELECTED REFERENCES - A Reading List

BOOKS:

Gengle, Dean (1984). The Netweaver's Sourcebook. Reading Massachusetts: Addison-Wesley.

Hiltz, R. and Turoff, M. (1978). The Network Nation. Reading Massachusetts: Addison-Wesley.

JOURNALS:

Byte, December 1984. Theme: Communications

Byte, December 1985. Theme: Computer Conferencing

RESEARCH ARTICLES:

Keiser, S. et al. (1985). Affect in computer-mediated communication: an experiment in synchronous terminal to terminal discussion. In Human-Computer Interaction. Hillsdale, N.J.: Lawrence Earlbaum Associates.

Krueger, G. and Chapanis, A. (1980) Conferencing and teleconferencing in three communications modes as a function of the number of conferees. Ergonomics, 23(2).

GENERAL INTEREST ARTICLES:

Van Gelder, L. (1985). The strange case of the electronic lover. MS., October.

Heuer, R. (1984). The ancient sourcetronaut. Link-up, August.

Milgram, S. (1984). Network love. Omni, October 7(1).

A SYSTEM TO EXPLORE:

People/Link
3215 North Frontage Road
Arlington Heights, IL 60004

Voice lines: 800-524-0100
312-870-5200 (Illinois)
Modem registration line: 800-826-5200

Online Contact: Bud Wellington
User ID: TEACH

MIDI - SIXTH GRADE GENERAL MUSIC
MARCHING TO A NEW DRUMMER

BY: Susan Young and Six Grade Music Students, Northbrook School District 28, Northbrook, IL

Our school district, #28 in Northbrook, Illinois, has fostered the introduction of the computer into all curricular areas. The interest has been not only in the "Basics", but also in the Fine Arts curriculum.

My first use of the computer in music classes was with some of the early Drill and Practice CAI's using the DAC board developed by David B. Williams at Illinois State University. Next, I began to use the Silver Burdett Courseware, Music FUNdamentals on the Apple computer. These three programs also involve Drill and Practice through the use of an animated figure named "Presto". This animated figure guides students through lessons involving note names, rhythms, note values, rest values, and bar lines. It is accessed through a plastic overlay provided with the software that simulates the piano keyboard. It is a very useful skill-based series and I still use it in grades two and up.

When our school district decided to implement the middle-school concept this year, it was decided that all sixth graders would be required to take one trimester (twelve weeks) of music. I was fortunate to be asked, not only to teach these sixth grade classes, but also to be responsible for curriculum development as well. The curriculum would consist of two parts:

1. A History of American Musical Theatre, Jazz, and "Pop" music from 1900 to the present.
2. An intensive study of electronic music with "hands-on" experience.

In the previous year, the district's music teachers, computer consultant, and Assistant Superintendent for Curriculum had discussed the possibility of purchasing a MIDI system for each attendance center at some time in the near future. Since we were wanting to offer the sixth graders something new and exciting in the middle school music program, the "future" arrived a little earlier than expected.

It was decided to purchase the Yamaha CX5M Music System. This included computer with MIDI-port, keyboard, printer, and various ROM cartridges with a disk drive. The cost was less than conversion to a similar system using our already in-use Apple computers.

MIDI (Musical Instrument Digital Interface) is essentially a link between those instruments capable of receiving and transmitting computer data to allow instruments of different manufacturers to work together as an efficient music-making system. In the early days of musical computers there were a variety of problems involved in interfacing one with another. For example:

1. Different ways of using voltage--some manufacturers increased voltage as pitch increased and some decreased voltage as pitch increased.

2. Differing clock rates (measured in pulses per quarter note (ppqn)); The Roland system (24ppqn); the Oberheim system (96 ppqn); the Fairlight CMI (384ppqn); etc.

The Oberheim system actually anticipated MIDI by devising a drum unit, polyphonic keyboard and sequencer that all worked with each other.

The advantages of using a MIDI system in an educational setting are many. The student has the opportunity to perform with a real instrument instead of a keyboard overlay on a computer keyboard. Technical proficiency is not a prerequisite for a quality experience and some students without very much "musical talent" can be very successful at this venture. There is the reward of instant playback of the student's work and also easy ways of editing the score. The student may also have the excitement of adding on new percussion and terrific rhythm parts in order to perfect the composition. The student may also choose to have a hard copy of the composition.

The implementation of the Electronic Music "hands-on" part of our trimester proved to be a very rewarding challenge. The first came when the system arrived in early October after school had already started. That meant playing a little catch-up for the teacher before the students would be allowed to start. I decided that the best thing would be to leave this unit for the last four weeks of the trimester.

With one system and 29 students in the largest class, the first challenge was in how to set up the process. I decided on the following method of work:

1. Figure out a way to have those at the system be able to use earphones while the rest of the class worked on other aspects of the unit.

2. Devise a tutorial, so that those working at the computer could get help from that source.

3. Split up into groups of 2 or 3 at a time so they could help each other should the group be having difficulty.

4. Construct a project that was within reach of ALL in the class, regardless of ability.

The figuring out how to set up the earphones took a little while, but it is now working like a charm through the CD input and the phones output on the stereo. At the other building, I have worked out a way to access earphones through one of the old, old phonographs that has a separate speaker. It means that the earphones will be in mono rather than the stereo output of the computer, but it works!

I decided to spend 2-3 days demonstrating the system to the sixth graders while they were following along in the tutorial and encouraging them to ask all kinds of questions about what they did not understand. I divided them into groups alphabetically and gave them their project. Each group was to work for 2 music periods. Each person was to enter 16 measures of a song that was already composed or that the group would compose itself. The 16 measures would be entered in one part and the next person would enter 16 measures in part two, etc. When finished, the parts would play together at the same time and sound well together. This worked very well, because I also had a check on each individual person's work.

What did not work well, and what I did not anticipate, was that most of the young people really had not digested the material from the demonstrations and most of them were not convinced that they should follow the directions as written in the tutorial. Many of the students did not go to the tutorial at all for help, and were floundering. It seemed that no

amount of cajoling certain students would make them aware that they should follow the directions step-by-step. This was very frustrating to me, and I decided that I would change my method of work for the next trimester. Despite this adversity, there were some surprisingly good projects turned in to me, and some students did come in for extra out-of-class help.

The second trimester, I decided that each group would give the rest of the class a demonstration (after mine) so that each group would have some "hands-on" experience before going out on their own. We spent 8 days taking the time to help each group and each student in that group get started, with the rest of the class observing, offering suggestions and trouble-shooting for the demonstration group. It was time-consuming, sometimes dull, and somewhat repetitive, but this group was much more ready to begin work and there were not near as many problems. It also seemed that if a certain group was having problems, there was a student who could help, which left me free to give the daily lesson to the other members of the class. I also put this unit at the beginning of the trimester and gave each group 3 times on the computer. This method is working much more smoothly than last. This teacher has really learned something!

What about the future of this program in District #28? It should be expanded to the point that we have at our middle school an entire music lab with enough CX5M's so that a whole class of groups may work at the same time. Which such a lab, the teacher will be free to circulate among the groups offering help and guidance. It will also offer the students more time on the system, in order that they may use some of the other features of the CX5M that are available. As it is now, they are only scratching the surface.

In the future, I would also hope to see this program expanded into the seventh and eighth grades either as a requirement or as a selective. With the addition of more systems, the inclusion of a voice sampler, drum machine, sequencer, and more sophisticated printer, there is no limit to where this program could go.

The students who were part of the presentation at the conference were all sixth graders. They were chosen in a random drawing from those who were interested in such an exciting adventure. Each had worked long hours preparing their presentation for the conference participants and were looking forward to such an unusual opportunity!

BIBLIOGRAPHY

TEXTBOOK: Electronic Music Marsh, Rinehart, and Savage
Macmillan Publishing Co., Inc. N.Y. 1980

COMPUTER SOFTWARE: Silver Burdett Courseware: Music FUNdamentals (Apple)
Beginning Music I: Reading and Playing Melody
Beginning Music II: Continuing Rhythm and Melody
Beginning Music III: Extending Rhythm and Melody Skills
(Very useful CAI program even as early as second grade)

AV MATERIALS: Filmstrip: How to Make Electronic Music (4 filmstrips and tape)
1. New sounds, new shapes
2. Changing sounds
3. Putting sounds together
4. Introduction to the synthesizer

EAV Company
Pleasantville, N.Y. 10570

Filmstrip: Pop/Rock Instruments: A Demonstration Ride
Michael Bennett

Pop Hits Publishing
3149 Southern Avenue
Memphis, Tenn. 38111

16mm Film: Discovering Electronic Music (Revised)
(Includes material about digital and voice synthesis)
Available through University of IL Film Center
(Reasonable rental fee)

BOOK: MIDI for Musicians Craig Anderton
Amsco Publishing, 1986
Division of Music Sales Corp.
24 E. 22nd St., New York, N.Y. 10010

PERIODICALS: Electronic Music Magazine
2608 Ninth Street
Berkeley, CA 94710

Music Educators Journal Dec., 1986: Vol. 73, #4
(Entire issue devoted to MIDI)
Publication of the Music Educator's National Conference
1902 Association Drive
Reston, VA 22091

LIST OF SPEAKERS

Peter Abrams
SASA
412 Joanne Lane
De Kalb, IL 60115

Jill Alexander
Our Lady of Mercy School
4416 N. Troy
Chicago, IL 60626

Linda Allison
CCSD #59
2123 S. Arlington Heights Rd.
Arlington Heights, IL 60005

Jay Amberg
Highland Park High School
433 Vine Avenue
Highland Park, IL 60035

Eric S. Anderson
Ohio Valley Area Libraries
St. Rte. 93
252 West Thirteenth
Wellston, OH 45692

Darlene Bandomir
Cherokee School
475 East Cherokee Road
Lake Forest, IL 60045

Rick Banducci
Friendship Junior High School
550 Elizabeth Lane
Des Plaines, IL 60018

Vickers Barrett
Meadowbrook School
Farm Road
Weston, MA 02193

Eileen Bell
Prospect High School
801 Kensington Road
Mt. Prospect, IL 60004

David Bergandine
University High
1212 W. Spring
Urbana, IL 61801

Karen Billings
Logo Computer Systems Inc.
121 Mt. Vernon St.
Boston, MA 02108

Robert Bortnick
CCSD #59
2123 S. Arlington Heights Road
Arlington Heights, IL 60005

Dean Brown
Picodyne Corporation
3000 Alpine Road
Portola Valley, CA 94025

Don Bunt
Chicago State University
116 Venetian Way
Orland Park, IL 60462

Paul Burton
DeSoto School District
P.O. Box 568
DeSoto, IL 62924

Maureen Cheever
Meadowbrook School
1600 Walters Avenue
Northbrook, IL 60062

Dennis Clodi
Educational Service Center #7
Box 69
Flossmoor, IL 60422

Ken Cull
Frost Junior High School
320 Wise Road
Schaumburg, IL 60193

Ann Dana
Hinsdale Junior High School
100 S. Garfield
Hinsdale, IL 60521

Allen Davis
Educational Service Center 2
19525 West Washington
Grayslake, IL 60030

SPEAKER LIST CONTINUED

Wally Dodge
New Trier High School
385 Winnetka Ave.
Winnetka, IL 60093

Tony Domenech
Teacher Support Software
P.O. Box 7130
Gainesville, FL 32605

Richard DuFour
Adlai E. Stevenson High School
16070 West Highway 22
Prairie View, IL 60069

Albert Erlebacher
Northwestern University
Kresge Hall
1859 Sheridan Road
Evanston, IL 60201

Beverly Fink
Des Plaines School District 62
2154 Elmira
Des Plaines, IL 60018

Frederick P. Frank
SASA
412 Joanne Lane
DeKalb, IL 60115

Stan Goldberg
Foundation Corporation
506 W. Armitage
Chicago, IL 60614

Bobby Goodson
Bobby Goodson & Associates
1666 Kitchener Drive
Sunnyvale, CA 94087

Brad Goral
New Trier High School
385 Winnetka Avenue
Winnetka, IL 60093

Mike Gordon
Westview Hills Middle School
630 - 65th Street
Clarendon Hills, IL 60514

Leonard Hall
Software Development Corp.
6350 Presidential Ct. SW
Suite 14
Ft. Myers, FL 33907

Steve Herring
Maine Township H.S. #207
1131 South Dee Road
Park Ridge, IL 60068

Robert Inglesby
Thomas Junior High School
303 E. Thomas Street
Arlington Heights, IL 60004

Jerry Justic
Specialized Data Systems
855 W. Prairie Ave.
Wheaton, IL 60187

Dennis Kapral
IBM Corporation
711 Jorie Blvd.
Oakbrook, IL 60521

T. D. Kimura
Washington University
Department of Computer Science
Campus Box 1045
St. Louis, MO 63130

Michael Land
National College of Education
2840 Sheridan Road
Evanston, IL 60201

Ann Landy
Penn State University
Dept. of Psychology
458 Moore Building
University Park, PA 16802

Judy Lange
Maine South H.S. Library
1111 S. Dee Road
Park Ridge, IL 60068

SPEAKER LIST CONTINUED

David Lowman
Tandy Corp./Radio Shack
1340 Remington Road
Schaumburg, IL 60173

Peter Lykos
Illinois Institute of Technology
Chicago, IL 60616

Muriel Mackett
SASA
412 Joanne Lane
DeKalb, IL 60115

Dale Manning
Maine Township H.S. District 207
1131 South Dee Road
Park Ridge, IL 60068

Anne Marshall
UNISYS
Chicago Government/Ed Bureau
1 Burroughs Center
Lombard, IL 60148

Margie Mason
Northern Illinois University
708 S. Second
DeKalb, IL 60115

John McConnell
Glenbrook South High School
4000 W. Lake Avenue
Glenview, IL 60025

John McDonald
Grayslake High School
400 N. Lake
Grayslake, IL 60030

Diane McGrath
398 Education
UN of IL
1310 S. 6th Street
Champaign, IL 61820

Dick McKenzie
IOICC/HORIZONS
217 East Monroe
Suite 203
Springfield, IL 62706

Jim Mecklenburger
National Schools Board Assn.
1680 Duke St.
Alexandria, VA 22314

Scott Meland
Deerpath Junior High
95 W. Deerpath
Lake Forest, IL 60045

Joe Meyer
Illinois Math & Science Academy
1500 West Sullivan Road
Aurora, IL 60506-1039

Mike Mills
Crystal Lake High School
45 W. Franklin Street
Crystal Lake, IL 60014

Rob Molek
Educational Service Center 10
105 Sage Street
Channahon, IL 60410

Mark Myers
Educational Service Center 15
P.O. Box 340
Charleston, IL 61920

Thomas Neudecker
Office of Academic Affairs
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213

Thomas O'Brien
Teacher's Center
Southern Illinois University
Edwardsville, IL 62025

Charlene Orr
Maine East High School
2601 Dempster
Park Ridge, IL 60068

Doris Peterson
Ideal Learning Inc.
327 So. Marshchall Road
Suite 200
Shakopee, MN 55379

SPEAKER LIST CONTINUED

Donald T. Piele
University of WI Parkside
Box 2000
Wood Road
Kenosha, WI 53141

Nancy Powell
Bloomington High School
School District 87
1202 E. Locust
Bloomington, IL 61701

Don Rawitsch
MECC
3490 Lexington Avenue N
St. Paul, MN 55126-8097

Sam Ritchie
Adlai E. Stevenson High School
16070 West Highway 22
Prairie View, IL 60069

M.D. Roblyer
Florida A & M University
College of Education
Tallahassee, FL 32307

Ted Sanders
Educational Service Center 4
421 N. County Farm Road
Wheaton, IL 60187

Terry Shevelenko
Video, Etc.
129 Skokie Valley Road
Highland Park, IL 60035

Dave Siden
Deerpath Junior High School
95 W. Deerpath
Lake Forest, IL 60045

Marty Siegel
University of Illinois
252 Engineering Research
103 S. Mathews Ave.
Urbana, IL 61801

Darrell Snedecor
Computer Systems Analyst
112 N. Gum Street
Harrisburg, IL 62946

Loren Sucher
MECC
3490 Lexington Avenue N
St. Paul, MN 55126-8097

Michael Tempel
Logo Computer Systems Inc.
555 W. 57th Street
Suite 1236
New York, NY 02108

Patrick W. Thompson
Mathematics Department
Illinois State University
Normal, IL 61761

Warren Toneman
Oak School
950 S. Oak
Hinsdale, IL 60521

Gary Tubb
UN of So. Florida-SVC 409
4202 E. Fowler Ave.
Tampa, FL 33620

Sandra Turner
National College of Education
2840 Sheridan Road
Evanston, IL 60201

Barbara T. Vincent
Meadowbrook School
Farm Road
Weston, MA 02193

Brad Waggoner
Educational Service Center 9
119 West Madison-Room 102
Downtown Courthouse
Ottawa, IL 61350

Mike Warner
Educational Service Center 4
421 N. County Farm Road
Wheaton, IL 60187

Doris Anne Weitermann
John Jay
1835 W. Pheasant Trail
Mt. Prospect, IL 60056

SPEAKER LIST CONTINUED

Bud Wellington
University of South Florida
SVC 409
4202 E. Fowler Avenue
Tampa, FL 33620

John Wick
Lagniappe Testing Inc.
953 Oak Drive
Glencoe, IL 60022

Susan Young
Northbrook SD #28
1475 Maple Avenue
Northbrook, IL 60062

VENDOR LIST PARTICIPANTS 1987

Robert K. Edwards
ALL AMERICAN SUPPLY & MAGNAVOX CORP.
1827 Waukegan Road
Glenview, IL 60025

Joan Flanagan
APPLE COMPUTER, INC.
8700 W. Bryn Mawr
Chicago, IL 60631

Randall V. Woerner
AT&T
15th Floor
333 West Wacker Drive
Chicago, IL 60606

Robert W. Brewster
BREWSTER SOFTWARE SYSTEMS
P.O. Box 19
Markville, MN 55048

John Ichkoff
CHICAGO V.I.C.E., INC.
444 Pacific Court
Wheeling, IL 60090

Carole Proebstle
DLM TEACHING RESOURCES
20756 N. Meadow Lane
Barrington, IL 60010

Barbara Neu
EDUCATIONAL ACTIVITIES, INC.
502 Redondo Drive - Unit 503
Downers Grove, IL 60516

Sandra Wahl
EDUCATIONAL RESOURCES
2354 Hassell Road
Hoffman Estates, IL 60195

Denise Boucher
FOLLETT SOFTWARE COMPANY
4506 Northwest Highway
Crystal Lake, IL 60014

Stan Goldberg
FOUNDATION CORPORATION
506 W. Armitage
Chicago, IL 60614

Bob Fowler
HARTLEY COURSEWARE
133 Bridge Street
Dimondale, MI 48821

David Howerton
HOUGHTON MIFFLIN COMPANY
Educational Software Division
125 E. Lake Street, Suite #304
Bloomingdale, IL 60108

Bruce Gjertsen
HRM SOFTWARE
175 Tompkins Avenue
Pleasantville, NY 10570

Dennis Kapral
IBM CORPORATION
Tower One
1701 Golf Road
Rolling Meadows, IL 60008

Mary Woolwine
INTELLECTUAL SOFTWARE-QUEUE
526 Boston Avenue
Bridgeport, CT 06610

Maisie Cheung
LOGO COMPUTER SYSTEMS, INC.
9960 Cote De Liesse
Lachine, Quebec
Canada H8T 1A1

Fay Paras
MIDWEST VISUAL EQUIPMENT CO.
6500 N. Hamlin Avenue
Chicago, IL 60645

Susan Hindman
MINDSCAPE, INC.
3444 Dundee Road
Northbrook, IL 60062

Nabih Mangoubi
NABIH'S, INC.
515 Davis
Evanston, IL 60201

Alyce Gambrell
NATIONAL COMPUTER SYSTEMS
4401 W. 76th Street
Minneapolis, MN 55435

Barbara Durlacher
NATIONAL SCHOOL SERVICES
632 S. Wheeling Road
Wheeling, IL 60090

Ronald G. Pace
PACEMARK TECHNOLOGIES, INC.
400 N. Franklin Street
Chicago, IL 60610

Jeffrey R. Walter
SCANTRON CORPORATION
1029 E. State Parkway
Schaumburg, IL 60173

Paul B. Harris
SCHOLASTIC, INC.
1940 Springer Road
Lombard, IL 60148

Ed Kozel
SOUTH-WESTERN PUBLISHING CO.
355 Conde Street
West Chicago, IL 60185

Kathy Hovelsrud
SPRINGBOARD SOFTWARE
7808 Creekridge Circle
Minneapolis, MN 55435

Hal Olsen
SRA
155 N. Wacker Drive
Chicago, IL 60606

Wendy Antini
SUNBURST COMMUNICATIONS
39 Washington
Pleasantville, NY 10570

David Lowman
TANDY CORP./RADIO SHACK
1340 Remington Road, Suite M
Schaumburg, IL 60173

Shelley Thomas
TEACHER SUPPORT SOFTWARE, INC.
P.O. Box 7130/2622
C-1 NW 43rd Street
Gainesville, FL 32605

Bruce Gjertsen
TECHNOLOGY GUIDANCE ASSOCIATES, INC.
610 S. Butterfield Road
Libertyville, IL 60048

Anne Marshall
UNISYS CORPORATION
One Burroughs Center - Suite 501B
Lombard, IL 60148

Terry Shevelenko
VIDEO ETC. - EDUCATION DIV.
129 Skokie Valley Road
Highland Park, IL 60035

R. J. Rustad
WICAT EDUCATION SYSTEMS
1020 31st Street
Suite 3000
Downers Grove, IL 60515

PROCEEDINGS INDEX

A

Abrams, Peter 66
Ackoff, Russell L. 94
Administration 67, 49, 199
Alexander, Jill 1
Allison, Linda 4
Anderson, Eric S. 10, 14
Appleworks 81

B

Bandomir, Darlene 18
Banducci, Richard J. 22
Bell, Eileen 25
Bergandine, David 28
Bortnick, Robert 31
Bulletin Board 74
Bunt, Don D. 35

C

CAD 22, 174
Career Information 110
Cheever, Maureen 38
Chemistry
    Secondary 28, 93
Communication 125, 244
Computers
    Literacy 76, 129
    Information Center 199
Contest
    Programming 163
    Problem Solving 154

D

Dalke, Deborah 125
Dana, Ann 42, 45
Data Bases 45
Davis, Allen 49
Dodge, Walter R.
    Figure 1 59
Dodge, Walter R. 52

E

Erlebacher, Albert 60

F

Fink, Beverly 63
Frank, Frederick 66
Funding 1

G

Ghostwriter 31
Gifted Education 98
Goodson, Bobby 70
Goral, Brad 74
Gordon, Michael 76
Graphics 163
Graphing 52

H

HORIZONS 110

I

Index 260
Industrial Arts 174
Information Systems 66
Inglesby, Bob 81

K

Kimura, Takayuki Dan 84
Kimura, T.D.
Figure 4.2 - Data Query 92
Kimura, T.D.
Figure 4.1 - Database Schema 91
Kimura, T.D.
Figure 3.3 - Terminating Iteration 90
Kimura, T.D.
Figure 3.2 - Infinite Iteration 90
Kimura, T.D.
Figure 3.1 - Fibonacci Number Program 89
Kimura, T.D.
Figure 2 - Factorial Function 88
Kimura, T.D.
Figure 1 - Completion Problems 87

L

Landy, Ann R. 125
Land, Michael L. 211
Lathrop, Ann 70

L

Library
- Automation 14
- Classification 148
- Online Applications 25
- Management 10

Logowriter 18
Lykos, Peter 93

M

Mackett, Muriel 66
Management
- Language Arts 61
- Mathematics 61
- Reading 61

Mathematics 60, 139, 180, 186, 211
McDonald, John C. 98
McGrath, Diane 101
McKenzie, Dick 110
Meland, Scott 114
Music 19, 114, 249

N

Nelson, Keith E. 125
Networking
- TOPS 172

Neudecker, Thomas 129

O

O'Brien, Thomas C. 139
Orr, Charlene 148

P

Physical Education
- Elementary 195

Piele, Donald T. 154
Polya, George 95
Powell, Nancy A. 163
Prince, George M. 95
Prinz, Philip M. 125
Problem Solving 93, 101, 139, 147, 223
Programming 84, 101, 163
- BASIC 163
- Elementary 63
- Logo 18, 78, 205, 211
- Secondary 35

R

Reading 4, 61, 125, 180
Research 38, 45, 49, 104, 167, 215, 237
Robotics 22, 205
Rubinstein, Moshe 95

S

Sanders, Ted 167, 172
Science 74, 93
    Elementary 42
Show and Tell 84
Siden, David 174
Siegel, Martin A.
    Figure 3. Darts 184
Siegel, Martin A.
    Figure 2. Corrective Feedback Paradigm 181
Siegel, Martin A.
    Figure 1. Flash Card Drill 181
Siegel, Martin A. 180
Social Science
    Elementary 46
Software
    Evaluation 70
    Music 114
    Problem Solving 143
    Science 42
Speaker List 253
Special Education 125

T

Technology
    Robotics and CAD 22
Thompson, Patrick W. 186
Thompson, Patrick W.
    Figure 1. and Figure 2. 187
    Figure 3. and Figure 4. 188
    Figure 5. and Figure 6. 189
    Figure 7. and Figure 8. 191
    Figure 9. and Figure 10. 192
Toneman, Warren 195
    Individual Physical Fitness Chart 198
Tubb, Gary W. 199, 205
Turner, Sandra 211

V

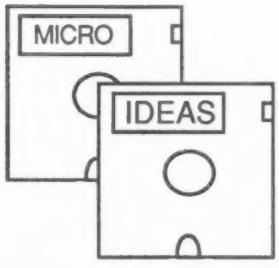
Vendor List 258
Videodisc
    Interactive 28, 125
Vincent, Barbara T. 223

W

Warner, Michael P. 172
Weitermann, Dorisanne 4
Wellington, Bud 237, 244
Wick, John W. 60
Word processing 18, 31, 77
Writing 4, 31, 38, 125
Writing To Read 4

Y

Yamaha CX5M Music System 249
Young, Susan 249



The Role of the Computer in Education VII
2701 Central Road
Glenview, IL 60025
312-998-5065